STOCHASTIC ESTIMATION METHODS IN GENERAL HIERARCHICAL MODELS

By

WOLFGANG S. JANK

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2001

To my family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. James Booth. Without his guidance and advice this work would have not been completed. Moreover, he encouraged me to continue a career in the academic field.

I would also like to thank Drs. Alan Agresti, James Hobert, Kirk Hatfield and Brett Presnell for serving on my committee and helping me with my research. Also, I want to thank all the faculty, staff and students, especially Bernhard, Brian, Galin, Jamie, Jeff, Patches and Ziyad, for their support and friendship.

Moreover, I want to thank my girlfriend Angel for bearing with me and supporting me, especially during the last months of my dissertation.

And finally, my deepest gratitude goes to my parents, Waltraud and Gerhard, and my sister, Sabina. Although they have not seen me very often during my last five years here in Florida, they never stopped supporting me with endless love and invaluable advice.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS iv

ABSTRACT viii

CHAPTERS

1 INTRODUCTION 1

1.1 Statistical Analysis of Clustered Data 1
1.2 Models for Normal Clustered Data 2
1.3 Models for Non-Normal Clustered Data 5
1.4 Maximum Likelihood Computation in Hierarchical Models . 8
1.5 Outline 13

2 MODEL AND LIKELIHOOD 15

2.1 The General Hierarchical Model (GHM) 15
2.1.1 The Generalized Linear Mixed model (GLMM) 16
2.1.2 The Linear Mixed Model (LMM) 21
2.1.3 The One-Way Balanced Mixed Model (OWMM) . . . 21
2.2 Maximum Likelihood Estimation 22
2.2.1 Logistic-Normal Model 22
2.2.2 Logistic-Normal Model with Multivariate Random Effects 23

3 DETERMINISTIC MAXIMUM LIKELIHOOD COMPUTATION . 24

3.1 Penalized Quasi-Likelihood 24
3.2 Quadrature 25
3.3 The Newton-Raphson (NR) Algorithm 26
3.4 The EM Algorithm 30
3.5 Analytical Comparison of Newton-Raphson and EM 35
3.6 Modifications of EM 38

4 STOCHASTIC MAXIMUM LIKELIHOOD COMPUTATION . . . 42

4.1 Introduction 42
4.2 Simulated Maximum Likelihood (SML) 44
4.3 Monte Carlo EM and Monte Carlo Newton-Raphson 47
4.3.1 Monte Carlo EM (MCEM) 47

4.3.2 Monte Carlo Newton-Raphson (MCNR) 52
4.4 Stochastic Approximation (SA) 57
4.4.1 Stochastic Approximation Newton-Raphson (SANR) . 59
4.4.2 Stochastic Approximation EM (SAEM) 64

5 EFFICIENCY OF MONTE CARLO EM AND SIMULATED MAXIMUM LIKELIHOOD 68

5.1 Efficiency of MCEM and SML in GLMMs 69
5.1.1 Asymptotic Monte Carlo Error 69
5.1.2 Discussion 71
5.1.3 Simulation Study 73
5.2 Two Examples 74
5.2.1 Mississippi River Data 75
5.2.2 McCulloch's Model 76
5.2.3 Practical Limitations of SML 77
5.2.4 More Efficient Use of MCEM 80

6 EFFICIENCY IMPROVEMENT WITH QUASI-MONTE CARLO 81

6.1 Quasi-Monte Carlo Integration 82
6.1.1 Low Discrepancy Sequences 84
6.1.2 Halton Sequences 85
6.1.3 Approximation Error of Quasi-Monte Carlo 86
6.1.4 Randomized Quasi-Monte Carlo 87
6.1.5 Randomized Halton Sequences 88
6.2 Quasi-Monte Carlo methods in GLMMs 89
6.2.1 Probability Integral Transformation for GLMMs . . . 89
6.2.2 Efficiency of SML using Quasi-Monte Carlo 90
6.3 Conclusion 93

7 EFFICIENCY OF STOCHASTIC APPROXIMATION 95

7.1 Convergence Rate of MCEM and SAEM 95
7.1.1 Univariate Parameter Case 100
7.1.2 Multivariate Parameter Case 101
7.2 Examples 104
7.2.1 Logistic-Normal Model 104
7.2.2 Poisson-Gamma Model 105
7.3 Efficiency of MCEM and SAEM 107
7.4 Conclusion 109

8 CONCLUSIONS 111

8.1 Summary of Results 111
8.2 Future Research 114

APPENDICES

A DERIVATIONS 116

A.1 Representation for the score function 116
A.2 Representation for the Hessian matrix 116
A.3 Derivations for the LMM 117
A.4 Estimating μ in the OWMM 119
A.5 Approximating $\boldsymbol{\tau}^2_{\text{MCEM}}$ and $\boldsymbol{\tau}^2_{\text{SML}}$ 120

B OX PROGRAM CODE 125

REFERENCES 127

BIOGRAPHICAL SKETCH 136

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

STOCHASTIC ESTIMATION METHODS IN GENERAL HIERARCHICAL MODELS

By

Wolfgang S. Jank

August 2001

Chair: James G. Booth
Major Department: Statistics

Hierarchical models are popular tools for the modeling of clustered data. A practical limitation to the use of these models is that the likelihood function typically involves an intractable, high dimensional integral. Several authors propose deterministic methods to approximate the intractable likelihood function. However, it has been shown that these methods do not always work well. As an alternative, stochastic estimation methods have been proposed. Stochastic estimation methods use simulation to approximate an intractable integral. Different stochastic estimation methods exist. In this dissertation we focus on five different methods: simulated maximum likelihood (SML), Monte Carlo EM (MCEM) and Monte Carlo Newton-Raphson (MCNR), as well as stochastic approximation EM (SAEM) and stochastic approximation Newton-Raphson (SANR).

First we describe the five individual methods and point out limitations and practical difficulties associated with each. In particular, we address the issue of

convergence, convergence rates, and stopping rules, as well as the choice of the starting values for each of these methods. Next, we perform analytical as well as empirical investigations to compare the efficiency of these methods. Specifically, we derive the asymptotic standard errors of MCEM and SML which show that in most practical applications of hierarchical models SML is very inefficient relative to MCEM. A simulation study and several examples support these analytical results. We also consider Quasi-Monte Carlo techniques in an attempt to increase the efficiency of SML. Then we characterize the convergence rate of MCEM and SAEM and show that in hierarchical models MCEM typically converges at a much faster rate. Our simulations support that this fast convergence rate can result in a much more efficient use of the total simulation amount for MCEM.

CHAPTER 1
INTRODUCTION

1.1 Statistical Analysis of Clustered Data

In many areas of research data arise in clusters or groups. For example in surveys or observational studies populations often have a natural hierarchical structure which leads to the collection of clustered data. Langford and Bentham (1997) study sudden infant death in England and Wales at district, county, and regional levels. Clearly, data collected from one district form a more homogeneous group than data from the remaining districts. This is also true, at the next level, for data from the same county. Similarly, Ramamurti (2000) uses data collected at firm-, industry- and country-levels to explain why emerging economies are privatizing state-owned enterprises, whereas Raudenbush et al. (1995) investigate psychological changes within married-couples based on data collected at the personal as well as the environmental level. Clustered data can also arise from taking measurements repeatedly on the same subject. In longitudinal studies, for example, data are collected on the same experimental unit over time or under multiple conditions leading to clusters of data from the same subject.

Statistical inference from data that occur in clusters is complicated by the fact that observations within each cluster are typically correlated. Simple statistical models assume independence between the observations and are therefore not appropriate for these types of data. For example, ANOVA or regression models require independent measurements and can lead to wrong conclusions if the independence assumption is violated.

Statistical models for correlated data, and software packages for implementing these models, are readily available when the data consist of clustered *normal* responses. Normal data typically occur when measurements are taken on a continuous scale (after applying a suitable transformation). However, when the data is non-normal, modeling, in particular model fitting, is complicated by computational difficulties. In the following we motivate the ideas behind models for normal and non-normal clustered data.

1.2 Models for Normal Clustered Data

Normal data that occur in clusters is often modeled using the linear mixed model (LMM). The LMM, which has been investigated extensively in Searle et al. (1992), extends the classical linear model by allowing the responses to be correlated. In particular, it assumes that the correlation among observations from the same cluster arises because there is a natural heterogeneity across clusters and that this heterogeneity can be represented as a probability distribution. To account for this heterogeneity, an unobserved random variable (a "random effect") from the probability distribution is incorporated additively into the model. Since observations from the same cluster share the same random effect, correlation is induced between the observations within the cluster. Estimates of the parameters are obtained by maximizing the marginal likelihood, which requires integrating the joint likelihood over the random effects. When the random effects distribution is assumed to be normal, the marginal likelihood for the LMM can be written in closed form, making estimation straightforward. Since software packages for fitting LMMs are readily available, these models are popular tools for clustered normal data in practice.

Table 1.1: Heights Data

Family Number, Gender and Height in Inches																	
1	F	67	1	F	66	1	F	64	1	M	71	1	M	72	2	F	63
2	F	63	2	F	67	2	M	69	2	M	68	2	M	70	3	F	63
3	M	64	4	F	67	4	F	66	4	M	67	4	M	67	4	M	69

Example. Table 1.1 is taken from the online documentation of the SAS [1] Institute (2000) and shows data on the heights (y_{ij}) of 18 individuals classified according to family i ($i = 1, 2, 3$ or 4) and gender j ($j = 1$ if gender=M or $j = 2$ if gender=F). A traditional two-way analysis of variance for this data results in a model of the form

$$y_{ijk} = \beta_0 + \beta_i^{\mathrm{F}} + \beta_j^{\mathrm{G}} + \beta_{ij}^{\mathrm{FG}} + \epsilon_{ijk}, \tag{1.1}$$

where β_i^{F} and β_j^{G} denote the main effects from family i and gender j and β_{ij}^{FG} the family $\times$ gender interaction effect. The error terms, ϵ_{ijk}, are assumed to be independent of each other and normally distributed with mean zero and variance σ_0^2.

Using SAS's procedure PROC GLM to fit model (1.1) to the data, we obtain, for example, for the estimate of the difference of the main gender effects

$$\hat{\beta}_1 - \hat{\beta}_2 = 1.1667 \text{ (StdErr=1.3229)}, \tag{1.2}$$

resulting in a P-Value of 0.3985.

Two of the important assumptions behind model (1.1) are that the data are normally distributed and independent. For the data in Table 1.1, the normality assumption is probably realistic since the data are observed heights. However,

[1] The official SAS homepage on the web (www.sas.com) explains that "Back in 1976, when SAS was founded, SAS was an acronym for statistical analysis system. But a lot has changed since 1976. [...] That is why today's SAS is no longer an acronym. [...] Rather, it is a brand name."

the assumption of independence is questionable. Since the data occur in clusters (families), it is very likely that observations from the same family are correlated, that is, not independent.

One way of introducing a correlation structure into model (1.1) is through the use of random effects. Using vector notation, we can write (1.1) in the form of the linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.3}$$

where $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_4)'$ is a suitably partitioned response vector with $\mathbf{y}_i$ containing the responses from cluster (family) i, $\boldsymbol{\beta}$ denotes the vector of fixed effects, $\mathbf{X}$ the corresponding design matrix and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$, where $\mathbf{I}$ is the identity matrix. The LMM extends (1.3) to

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}, \tag{1.4}$$

where the random effect $\mathbf{u}$ is assumed to be normally distributed with mean zero and covariance matrix $\mathbf{G}$, independent of $\boldsymbol{\epsilon}$, and $\mathbf{Z}$ denotes the design matrix for $\mathbf{u}$.

Estimation in the LMM is straightforward. Notice that equation (1.4) implies that marginally,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}^*, \tag{1.5}$$

where $\boldsymbol{\epsilon}^* \sim \mathcal{N}(\mathbf{0}, \mathbf{V})$ and $\mathbf{V} = \sigma_0^2\mathbf{I} + \mathbf{Z}\mathbf{G}\mathbf{Z}'$. It follows from equation (1.5) that the marginal likelihood for the LMM is available in closed form and thus estimates for $\boldsymbol{\beta}$, σ_0^2 and $\mathbf{G}$ can be obtained with little effort (using, e.g., the EM or Newton-Raphson algorithms described in Chapter 3).

The LMM also allows for a flexible correlation structure among the responses. For example, if $\mathbf{V}$ is of block-diagonal form, $\mathbf{V} = \text{BlockDiag}(\mathbf{V}_1, \ldots, \mathbf{V}_4)$, such that $\text{Cov}(\mathbf{y}_i, \mathbf{y}_i) = \mathbf{V}_i$ and $\text{Cov}(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{0}$ $(i \neq j)$, then observations within the same cluster are correlated whereas observations between different clusters are independent.

A simple modification of model (1.1) accounts for the clustering of the data in Table 1.1. Declaring the main family effect as random sets up a common correlation among all observations having the same level of family. Furthermore, declaring the family $\times$ gender interaction effect as random models an additional correlation between all observations that have the same level of both family and gender. One interpretation of this effect is that a female in a certain family exhibits more correlation with the other female in that family than with other males, and likewise for males. This results in the model

$$y_{ij} = \beta_0 + \beta_j + u_i + u_{ij} + \epsilon_{ij}, \tag{1.6}$$

where the u_i's and u_{ij}'s are a random sample from a standard normal distribution with mean zero and variances σ_1^2 and σ_2^2, respectively.

Using SAS's procedure PROC MIXED to fit model (1.6) to the data, the estimated difference of the gender effects becomes

$$\hat{\beta}_1 - \hat{\beta}_2 = 3.3621 \text{ (StdErr=1.1923)}, \tag{1.7}$$

resulting in a P-Value = 0.0667, which is quite different from the estimate in (1.2).

1.3 Models for Non-Normal Clustered Data

Linear mixed models are useful tools for modeling normal clustered data. However, in many practical situations the responses are not normally distributed, making the use of these models inappropriate. Hierarchical models extend the ideas of LMMs to more general types of data, including non-normal responses. One example of a hierarchical model that currently receives a great deal of attention in the literature is the generalized linear mixed model (GLMM) (Gilmour et al., 1985; Breslow and Clayton, 1993; McCulloch, 1994, 1997). The GLMM extends the generalized linear model (GLM) (Nelder and Wedderburn, 1972; McCullagh and Nelder, 1989) and contains both, the LMM and the GLM, as a special case.

The GLM is a natural extension of the classical linear model allowing for a larger class of response distributions. Specification of a GLM consists of three parts: a random component, a systematic component and a link function. The random component specifies the distribution of the responses; typically, the y_i's are assumed to be independent with a distribution in the exponential family (e.g. normal, binomial, Poisson, gamma, etc.). The systematic component is a linear function of the covariates

$$\eta_i = \mathbf{x}_i'\boldsymbol{\beta}, \tag{1.8}$$

where η_i is called the linear predictor. The link function $g(\cdot)$ is a monotonic differentiable function which relates the mean response to the linear predictor

$$\eta_i = g(E[y_i]). \tag{1.9}$$

When the response distribution is normal and the link is the identity function, the GLM reduces to the classical linear model.

A limitation of GLMs is that they assume independent responses. Just as modifications of classical linear models are used to analyze correlated normal responses, modifications of the GLM can handle non-normal outcomes. The GLMM extends the GLM to correlated responses through the use of random effects. In a GLMM one includes random effects additively into the linear predictor and assumes that, conditional on the random effects, the response follows a GLM. Specification of a GLMM is completed by assuming a marginal distribution for the random effects (see Section 2.1.1 for a more detailed description of the GLMM).

Although hierarchical models have great intuitive appeal, their practical use is often limited by computational difficulties. For example in the GLMM, the assumption of normally distributed random effects typically leads to an analytically intractable likelihood, making maximum likelihood estimation complicated.

Table 1.2: Weil Data

Treatment, liter number and litter size after 4 and 21 days															
c	1	13	13	c	2	12	12	c	3	9	9	c	4	9	9
c	5	8	8	c	6	8	8	c	7	13	12	c	8	12	11
c	9	10	9	c	10	10	9	c	11	9	8	c	12	13	11
c	13	5	4	c	14	7	5	c	15	10	7	c	16	10	7
t	1	12	12	t	2	11	11	t	3	10	10	t	4	9	9
t	5	11	10	t	6	10	9	t	7	10	9	t	8	9	8
t	9	9	8	t	10	5	4	t	11	9	7	t	12	7	4
t	13	10	5	t	14	6	3	t	15	10	3	t	16	7	0

Example. Table 1.2 shows data from Weil (1970), also studied by McCulloch (1994). In this experiment 16 pregnant rats received a control diet (c) and 16 received a chemically treated diet (t). The size for the resulting 32 litters was recorded after 4 and 21 days.

In order to model the survival time, y_{ijk}, of a newborn rat k in litter j receiving treatment i, McCulloch (1994) assumed a model of the form

$$y_{ijk} = \beta_i + u_{ij} + \epsilon_{ijk}, \tag{1.10}$$

where β_i is the treatment mean, u_{ij} a random effect due to litter j receiving treatment i, and ϵ_{ijk} residual error. He assumed further that the u_{ij}'s and the ϵ_{ijk}'s are random samples from a normal distribution with mean zero and variances σ_1^2 and 1, respectively. Notice that the inclusion of the random effect u_{ij} into the model allows the survival times of two newborn rats from the same litter to be correlated with each other.

Notice, however, that instead of observing the survival times y_{ijk} directly, the researcher only observes whether or not rat k from litter j under treatment i dies between day 4 and 21. That is, he only observes a Bernoulli response, say w_{ijk}, where

$$w_{ijk} = \begin{cases} 0 & \text{if } y_{ijk} \leq \gamma \\ 1 & \text{if } y_{ijk} > \gamma \end{cases}, \tag{1.11}$$

and γ is a threshold value. Without loss of generality, assume $\gamma = 0$. If $\Phi(\cdot)$ denotes the standard normal cdf; then equation (1.10) implies that conditional on u_{ij},

$$\pi_{ij} \equiv \text{Prob}(w_{ijk} = 1|u_{ij}) = \Phi(\beta_i + u_{ij}). \tag{1.12}$$

Notice that, conditional on u_{ij}, the model in (1.10) and (1.11) defines a GLM with independent Bernoulli responses w_{ijk}, the linear predictor $\eta_{ijk} \equiv \beta_i + u_{ij}$ and the function $g(\cdot) \equiv \Phi^{-1}(\cdot)$ which links the linear predictor to the (conditional) mean, $E[w_{ijk}|u_{ij}] = \pi_{ij}$. The assumption of normally distributed random effects, u_{ij}, completes the specification of the GLMM.

Maximum likelihood estimation for this model, however, is complicated by the fact that the likelihood function is analytically intractable. If $x_{ij} = \sum_k w_{ijk}$ and if m_{ij} denotes the litter size after day 4, then, conditional on u_{ij}, $x_{ij}|u_{ij} \sim$ Binomial(π_{ij}, m_{ij}). Let $\mathbf{w}$ denote the vector of observed responses and $\mathbf{u}$ the vector of random effects. The (marginal) likelihood of the observed data is obtained by integrating out the random effects in the joint likelihood of $\mathbf{w}$ and $\mathbf{u}$,

$$L(\boldsymbol{\beta}, \sigma_1^2|\mathbf{w}) \propto \prod_{i,j} \int \pi_{ij}^{x_{ij}} (1 - \pi_{ij})^{m_{ij} - x_{ij}} \varphi(u_{ij}/\sigma_1) du_{ij}, \tag{1.13}$$

where $\varphi(\cdot)$ denotes the standard normal pdf. The maximum likelihood estimates of $\boldsymbol{\beta}$ and σ_1^2 are obtained by maximizing $L(\boldsymbol{\beta}, \sigma_1^2|\mathbf{w})$. However, estimation is not straightforward, since the integral in (1.13) does not have a closed form solution.

1.4 Maximum Likelihood Computation in Hierarchical Models

Since maximum likelihood estimation in hierarchical models is complicated by the fact that the likelihood function typically involves an analytically intractable integral, much of the work in this area has focused on computational aspects. In the following we review different approaches in the literature for maximum likelihood computation.

In the early literature, Williams (1982) proposed an iterative scheme based on a quasi-likelihood approach for estimating the parameters in a GLMM. His approach, however, only worked well when the variability of the random effects was small.

Anderson and Aitkin (1985) suggested the use of numerical integration to approximate the intractable integral. In particular, they suggested using Gauss-Hermite quadrature within the context of an EM algorithm to approximate the mode of the likelihood. Liu and Pierce (1994) and Pinheiro and Bates (1995) suggested using quadrature methods to approximate the likelihood directly. However, a major disadvantage of numerical integration is that the error of approximation increases with the dimension of the integral (e.g. Evans and Swartz, 1995). It is therefore not recommended for multivariate random effects with a complex correlation structure. Moreover, establishing exactly how the error (in the approximation of the integral) propagates through the maximization can be difficult (Crouch and Spiegelman, 1990).

Several authors proposed to approximate the intractable likelihood function analytically. The model fitting algorithms of Breslow and Clayton (1993) and Wolfinger and O'Connell (1993) are essentially derived from a Laplace approximation to the intractable likelihood integral. Their algorithms involve iterative fitting of LMMs to the data and work well for observations that can be approximated well by a normal distribution (e.g., Binomial data with large cell counts). However, recent research has shown that these pseudo-likelihood estimates are substantially biased in some circumstances (Kuk, 1995; Breslow and Lin, 1995; Lin and Breslow, 1996; Jiang, 1998).

There are several approaches in the literature to choose the random effects distribution such that the likelihood is available in closed form. A number of such models (e.g., the beta-binomial or the Poisson-gamma model) are found in the

literature on over-dispersion (Stein, 1988; Conaway, 1990; Hinde and Demetrio, 1998). A common criticism of this approach is that distributions chosen in this way do not allow for modeling complex interdependence between multivariate random effects.

The choice of the random effects distribution can influence the parameter estimates of the fixed effects (see Heckman and Singer, 1984; Neuhaus et al., 1992). This has led to recent work focusing on non-parametric approaches for fitting hierarchical models (Follmann and Lambert, 1989; Aitkin, 1996; Friedl, 1998; Aitkin, 1999). Such approaches assume that the random effects follow a discrete distribution with unknown masses and mass points. However, Aitkin (1999) and Heckman and Singer (1984) noted that this approach is primarily useful when the random effects distribution is a nuisance rather than of direct interest, since the non-parametric estimate of that distribution may be poor even for large samples.

The focus of this dissertation is on stochastic estimation methods. Stochastic estimation methods are computer intensive techniques that have become popular with the development of powerful computing facilities. Stochastic estimation methods use simulation to approximate, for example, intractable integrals. In the field of statistics, integrals can often be represented as an expectation of the form

$$E[h(U)] = \int h(u)f(u)du, \tag{1.14}$$

for a random variable U with density $f(\cdot)$. If one can generate (or *simulate*) a sample $u_1, \ldots, u_M$ from f, then the integral in (1.14) can be approximated by the empirical average

$$\frac{1}{M}\sum_{k=1}^{M} h(u_k). \tag{1.15}$$

This approach is often referred to as Monte Carlo integration. An excellent introduction into Monte Carlo methods can be found in Robert and Casella (1999).

Several authors have suggested simulation to approximate the value of the likelihood in hierarchical models (Geyer and Thompson, 1992; Gelfand and Carlin, 1993; McCulloch, 1997). They propose to estimate the intractable likelihood integral by an empirical average, similar to (1.15), and obtain the parameter estimate by numerically maximizing this estimate. This approach can often be found in the literature under the name of simulated maximum likelihood (SML). SML is especially popular in economics where much of the work on this topic has been done (Lee, 1992, 1995, 1998, 1999; Gouriéroux and Monfort, 1993; Moon and Stotsky, 1993; Danielsson, 1994; Crepon and Duguet, 1997; Liesenfeld, 1998).

An alternative to approximating the likelihood function directly is to compute only its maximum. The EM algorithm (Dempster et al., 1977; Wu, 1983) iterates between expectation and maximization steps and produces a sequence of estimates that converges to a (local) maximum of the likelihood. In hierarchical models, however, the expectation step is typically not available in closed form. This has led to the development of the Monte Carlo EM (MCEM) algorithm (Wei and Tanner, 1990; McCulloch, 1994, 1997; Chan and Ledolter, 1995; Booth and Hobert, 1999; Natarajan et al., 2000), in which the intractable expectation is replaced by a Monte Carlo approximation. One difficulty associated with MCEM is that it does not converge unless one increases the Monte Carlo sample size, M, as the algorithm progresses.

Another popular method for computing the maximum of a function is the Newton-Raphson algorithm. Given a current parameter estimate, Newton-Raphson uses the score function and the Hessian matrix to compute the next parameter update. However, since the score function is defined as the first derivative of the log-likelihood, in hierarchical models Newton-Raphson typically involves evaluation of an intractable integral. The Monte Carlo Newton-Raphson (MCNR) algorithm (Kuk and Cheng, 1997; McCulloch, 1997; Gauderman and Navidi, 2001) replaces

this integral by a Monte Carlo approximation. Similar to MCEM, MCNR requires M to be increased successively for convergence.

A question that typically remains unanswered for both algorithms, MCEM and MCNR, is *how* M should be increased. Wei and Tanner (1990) noted that it is inefficient to start with a large Monte Carlo sample size when the current parameter update is far from the maximizer of the likelihood. They went on to suggest that M should be increased with the number of iterations but did not say exactly how this should be done. Many authors (including Wei and Tanner) suggest that the decision to increase M should be made after inspecting a plot of the parameter updates versus the iteration number. Clearly, this decision becomes more difficult when the dimension of the parameter increases. Moreover, it does not provide a way for automatically increasing M. Booth and Hobert (1999) are the first to develop an automated MCEM algorithm. They suggested to increase the Monte Carlo sample size when the current parameter update is swamped by Monte Carlo error. Their argument for increasing M, however, is valid only for independent samples, $u_1, \ldots, u_M$. In situations in which it is not possible to generate independent samples, the question about the optimal choice of M still remains.

On the other hand, there exist versions of the above two algorithms that converge with constant M. These versions are based on the ideas of stochastic approximation (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952). In particular, Delyon et al. (1999) consider a stochastic approximation version of the EM (SAEM) algorithm. Similarly, Ruppert (1985) and Gu and Li (1998) discuss stochastic approximation versions of Newton-Raphson (SANR). The appeal of these versions is that they do not require any additional effort to decide whether and when to increase M, which, at first glance, appears to be an advantage over MCEM

or MCNR. There are, however, disadvantages associated with SAEM and SANR, as we will explain later in this work.

Most of the literature on stochastic estimation methods, with the exception of McCulloch (1997), focuses on one individual method only, making no attempt to compare competing approaches. McCulloch (1997, p.165) compares SML, MCEM and MCNR empirically. He finds in simulations that "SML performs poorly [...]. SML is much slower than either MCEM or MCNR, but it converges to a value much further from the MLE." McCulloch's investigations are certainly a first step towards determining the most suitable estimation method for hierarchical models. However, his findings are of empirical nature only. Moreover, they do not include SAEM and SANR, making broader and more theoretical investigations desirable.

1.5 Outline

The purpose of this dissertation is to compare the performance of stochastic estimation methods. We conduct empirical as well as analytical investigations to compare the efficiency of SML, MCEM, MCNR, SAEM and SANR. This dissertation is organized as follows.

In Chapter 2 we describe the general hierarchical model and its wide range of applicability. We describe some important special cases of this model and illustrate them on several examples. Then we discuss maximum likelihood estimation and problems associated with it. We present the marginal likelihood function and argue that it typically involves an analytically intractable integral.

Chapter 3 concentrates on deterministic methods to compute the intractable likelihood. First we introduce the ideas behind numerical integration and analytical approximation to the intractable integral and point out limitations to these two approaches. Then we discuss Newton-Raphson and the EM algorithm. We outline the ideas of the two algorithms and illustrate them on several examples. We

compare the two algorithms analytically and conclude this chapter by describing important modifications of EM.

In Chapter 4 we consider stochastic methods to compute the intractable likelihood. First we introduce the general ideas of Monte Carlo integration. Then we describe SML and point out difficulties associated with it. We proceed by discussing MCEM and MCNR. We illustrate both algorithms on several examples, discuss convergence and the choice of the starting values. Then we introduce the general ideas of stochastic approximation and describe SAEM and SANR. We use several examples to illustrate both algorithms and discuss the difference to their Monte Carlo versions. We conclude by addressing convergence rates and stopping rules for stochastic approximation algorithms.

Chapter 5 concentrates on the comparison of MCEM and SML. We derive the asymptotic Monte Carlo standard errors of MCEM and SML analytically and use this result to investigate the efficiency of these two methods. We conduct a simulation study and illustrate our findings on several examples.

Chapter 6 focuses on efficiency improvement using Quasi-Monte Carlo. First we describe the ideas of Quasi-Monte Carlo and its difference to classical Monte Carlo. Then we demonstrate how to apply Quasi-Monte Carlo to hierarchical models and conduct a simulation study to investigate the efficiency improvement of SML using Quasi-Monte Carlo over classical Monte Carlo.

Chapter 7 concentrates on the comparison of MCEM and SAEM. We characterize the convergence rate of MCEM and SAEM in the univariate as well as in the multivariate parameter case. We use this result to investigate the efficiency of these two methods. We conduct a simulation study and use several examples to illustrate our results.

The dissertation concludes with a summary of the most important findings and a discussion of future research topics (Chapter 8).

CHAPTER 2
MODEL AND LIKELIHOOD

2.1 The General Hierarchical Model (GHM)

Simple statistical models often assume independence among the observations. For example, logistic regression, which is useful for modeling binomial data, requires the responses to be independent of each other. In many applications, however, the dependence structure of the data is more complex. In particular, observations are often clustered, with observations within clusters being correlated. In the social sciences, for example, the performance of students within the same school or school district tend to be positively correlated. A medical researcher will typically find that the responses of patients to a treatment are similar within the same treatment-center. In economics, unemployment rates tend to be correlated within the same region. More generally, if repeated observations are taken within a sampling unit then they are typically (positively) correlated.

Hierarchical models are popular tools that enable the researcher to account for correlation within a set of observations. A recent introduction to hierarchical modeling from a statistical point of view can be found in Hobert (2000). Following his review, our starting point will be the general two-stage general hierarchical model (GHM).

Let $\mathbf{y} = (y_1, \ldots, y_n)$ be an n-vector of responses and let $\mathbf{u} = (u_1, \ldots, u_q)$ be a q-vector of *unobserved* random effects or missing data. Let $\boldsymbol{\psi} = (\boldsymbol{\psi}_1', \boldsymbol{\psi}_2')'$ be a S-vector of unknown parameters. At the first stage of the hierarchy the conditional density of $\mathbf{y}$ given $\mathbf{u}$ is assumed to be of a specific parametric form, $f(\mathbf{y}|\mathbf{u}; \boldsymbol{\psi}_1)$. Then, at the second stage of the hierarchy, the marginal density of $\mathbf{u}$, denoted by

$g(\mathbf{u};\boldsymbol{\psi}_2)$, is specified. Let $f(\mathbf{y},\mathbf{u};\boldsymbol{\psi}) = f(\mathbf{y}|\mathbf{u};\boldsymbol{\psi}_1)g(\mathbf{u};\boldsymbol{\psi}_2)$ be the *joint* density of the observed and unobserved data and let $f(\mathbf{y};\boldsymbol{\psi})$ denote the marginal density of the responses.

In the following we discuss several special cases of the GHM. These are the following: the one-way balanced mixed model (OWMM), the linear mixed model (LMM) and the generalized linear mixed model (GLMM). Using the notation "A $\subseteq$ B" to express that model A is a special case of model B, the ordering of these models is

$$\text{OWMM} \subseteq \text{LMM} \subseteq \text{GLMM} \subseteq \text{GHM}.$$

2.1.1 The Generalized Linear Mixed model (GLMM)

A special case of the GHM which currently receives a great deal of attention is the GLMM (Gilmour et al., 1985; Breslow and Clayton, 1993; McCulloch, 1994, 1997). The GLMM extends the generalized linear model (GLM) by allowing for additional components of variability due to unobservable random effects.

Specifically, following Booth and Hobert (1999), we let $\mathbf{x}_i$ and $\mathbf{z}_i$ be p- and q-vectors of covariates associated with the ith response y_i, $i = 1,\ldots,n$. Let $\mathbf{X}$ and $\mathbf{Z}$ be the corresponding $(n \times p)$ and $(n \times q)$ model matrices. We assume that, conditional on $\mathbf{u}$, the data $\mathbf{y}$ arise from a GLM with linear predictors, $\eta_i = \mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}_i'\mathbf{u}$, where $\boldsymbol{\beta}$ is a p-vector of unknown regression coefficients. Writing $\mu_i = E(y_i|\mathbf{u})$ for the conditional mean of the ith response, we assume that μ_i and the linear predictor η_i satisfy $g(\mu_i) = \eta_i$, where the link function g may be any monotonic differentiable function. The responses are assumed to be conditionally independent, stemming from an exponential family

$$f(y_i|\mathbf{u};\boldsymbol{\beta},\sigma_0^2) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\sigma_0^2)} + c(y_i,\sigma_0^2)\right\}. \tag{2.1}$$

The mean and the canonical parameter are related through the equation $\mu_i = b'(\theta_i)$ (see McCullagh and Nelder, 1989, Chapter 2). The likelihood function for the

conditional generalized linear model, in which the effects vector is treated as fixed but unknown parameter, is given by

$$f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta},\sigma_0^2) = \prod_{i=1}^{n} f(y_i|\mathbf{u},\boldsymbol{\beta},\sigma_0^2).$$

Specification of the model is completed by assuming that $\mathbf{u}$ is a q-variate random variable with a parametric density, $g(\mathbf{u};\boldsymbol{\psi}_2)$.

In a GLMM one typically assumes that $\mathbf{u}$ is a mean-zero multivariate *normal* random variable with covariance matrix $\mathbf{G} = \mathbf{G}(\boldsymbol{\psi}_2)$, where $\boldsymbol{\psi}_2$ denotes a vector of variance components. (To emphasize that $\boldsymbol{\psi}_2$ is a vector of variance components many authors write $\boldsymbol{\sigma}_1^2$ in place of $\boldsymbol{\psi}_2$. We will adopt this notation in the following chapters). However, other distributions for $\mathbf{u}$ are possible. In particular, Lee and Nelder (1996) consider distributions *conjugate* to that of the response. This approach leads to the hierarchical generalized linear model (HGLM).

The following examples are special cases of the GLMM and HGLM. First, we present an example of a GLMM that is often found useful to model clustered binary data.

Logistic-Normal Model. Hierarchical models are popular tools for educational data (see, e.g., Woodhouse et al., 1996; Raudenbush and Bryk, 1986). For example, suppose that in a study of factors that affect school performance, data are collected from m randomly selected schools. The m schools might be a random sample from the collection of all schools in the county, region or school district. Within each school, r students are randomly selected and their performance (S=successful or U=unsuccessful) over the last school year is recorded (see Figure 2.1). Thus there is a binary response, y_{ij}, for student j ($j = 1,\ldots,r$) in school i ($i = 1,\ldots,m$), where $y_{ij} = 1$ or 0 depending on whether the student is successful or unsuccessful, respectively. Suppose further that there is a vector of explanatory variables (such as socioeconomic factors) associated with each student. If the researcher is

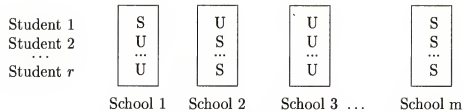

Figure 2.1: Student school performance

interested in modeling the probability of a successful student performance, logistic regression is a natural model choice. However, observations *within* the same school are often correlated due to unobserved effects like, for example, school equipment, the teaching method or the teacher performance.

The logistic-normal model provides a flexible way of accounting for correlation between subjects within the same group. It models the logit of the probability of success for subject j in group i as

$$\text{logit}(\pi_{ij}) = \mathbf{x}'_{ij}\boldsymbol{\beta} + u_i, \tag{2.2}$$

where $\text{logit}(\pi_{ij}) = \log(\pi_{ij}/(1-\pi_{ij}))$ and $\pi_{ij} = \text{Prob}(y_{ij} = 1|u_i)$. Conditional on u_i, the responses are assumed to be independent Bernoulli(π_{ij}). Specification of the logistic-normal model is completed by assuming that $u_i \sim N(0, \sigma_1^2)$. The inclusion of a random school effect, u_i, induces a positive correlation between the responses for the students in the same school.

Clearly, the logistic-normal model is a special case of the GLMM. It uses the logit-link function to model Bernoulli responses and assumes normally distributed random effects. Let us now focus our attention on count data. The modeling of count data is often complicated by the presence of over-dispersion. In the next paragraph we present an example of a GLMM that is useful to model over-dispersed count data.

Poisson-Normal Model. The following example is taken from Agresti et al. (2001). They consider data from the 1990 General Social Survey, where one of

the questions asked was "Within the past 12 months, how many people have you known personally that were victims of homicide?" One natural model for count data of this kind is a Poisson regression model. A severe limitation of the Poisson model is that the variance must be identical to the mean. However, count data often show over-dispersion, with the variance exceeding the mean. Indeed, Agresti et al. (2001) reported that the mean response for blacks was .522 with a variance of 1.150; for whites the mean was .092 with a variance of .155. The ratio of the variance to the mean for each race provides evidence of over-dispersed data for the Poisson model.

Generalized linear mixed models provide a flexible way of accounting for over-dispersion of count data with respect to the Poisson distribution. Specifically, assume that the distribution of the data is Poisson; that is, one models the log of the mean response of subject j in (ethnicity-) group i as

$$\log(\mu_{ij}) = \mathbf{x}'_{ij}\boldsymbol{\beta} + u_{ij}, \tag{2.3}$$

where, conditionally on u_i, response y_{ij} is assumed to stem from a Poisson(μ_{ij})-distribution. Assuming that $u_i \sim N(0, \sigma_1^2)$ leads to the Poisson-normal model. The inclusion of the random effect, u_i, typically accounts for some of the extra variability in the data.

The previous two examples were special cases of the GLMM and focused on normally distributed random effects. However, other distributional assumptions are possible. In particular, changing the random effects distribution in the previous example from normal to one which is conjugate to the Poisson distribution leads to an example of a HGLM.

Poisson-Gamma Model. Consider the previous example. If we assume that the random effects, u_i, vary according to a gamma distribution, we obtain the Poisson-gamma model. Since the gamma distribution is conjugate to the Poisson

Table 2.1: Pregnancy rates for women under 18 in 13 North Central Florida counties over the three-year period 1989-1991. (Gainesville Sun, April 30, 1994)

275	50	110	104	21	8	41	7	30	243	129	38	22
8544	1032	4851	2064	480	399	513	198	1050	8259	2946	1053	405

1^{st} row: number of births to women under 18 (y_i)
2^{nd} row: total number of births (n_i)

distribution (and since we assume the log-link in (2.3) for the mean), the likelihood function (discussed later in Section 2.2) is analytically tractable. However, a criticism of this approach is that the gamma distribution does not allow for modeling complex interdependence between multivariate random effects, which is often a severe limitation to HGLMs.

Beta-Binomial Model. The data in Table 2.1 concerns pregnancy rates for girls under 18 in 13 North Central Florida counties. Booth and Caffo (2001) report that the empirical variability in the child pregnancy rates among the counties is far greater than the binomial sampling variability with a common underlying rate (deviance = 89.86 with df=12). One approach to account for the extra variability in the data is to use the logistic-normal model described earlier. However, instead of assuming a normal random effects distribution one can alternatively use a beta distribution. Notice that the beta distribution is conjugate to the binomial distribution. This approach leads to the beta-binomial model in which the child pregnancy rates vary randomly among the counties according to a beta distribution; that is, conditional on u_i, $i = 1, \ldots, 13$, one assumes that

$$y_i | u_i \sim \text{Binomial}(n_i, u_i), \tag{2.4}$$

where $u_i \sim \text{Beta}(\alpha, \beta)$.

2.1.2 The Linear Mixed Model (LMM)

The linear mixed model is a special case of the GLMM, which uses the identity link function and assumes normally distributed responses. In this case the response y_i is a linear combination of fixed and random effects and an error term ϵ_i,

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}_i'\mathbf{u} + \epsilon_i.$$

The vector of errors, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)'$, follows a multivariate normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{W}^{-1})$, with $\mathbf{W} = (1/\sigma_0^2)\mathbf{I}$, and is uncorrelated with the vector of random effects, $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{G})$.

The LMM has been discussed extensively in the literature (see, e.g., Searle et al., 1992). It is an important special case of the GHM, since many techniques, developed for LMMs, have carried over to the theory of GLMMs and GHMs. For example, the model-fitting algorithms of Breslow and Clayton (1993) and Wolfinger and O'Connell (1993) developed for GLMMs (and discussed later in Chapter 3) involve iterative fitting of LMMs.

2.1.3 The One-Way Balanced Mixed Model (OWMM)

The one-way balanced mixed model is one of the simplest mixed models and is a special case of the LMM. Assume that a continuous response, y_{ij}, has been observed on subject j in group i, where $i = 1, \ldots, m$ and $j = 1, \ldots, r$. We model y_{ij} as

$$y_{ij} = \mu + u_i + \epsilon_{ij}, \tag{2.5}$$

where μ is the overall mean, u_i the effect of group i, and ϵ_{ij} the error associated with the jth response in group i. The u_i's and ϵ_{ij}'s are assumed to be independent random samples from $N(0, \sigma_1^2)$ and $N(0, \sigma_0^2)$, respectively. Later in this work we consider the case where the variance components, σ_0^2 and σ_1^2, are known and we are interested in estimating only μ. Although this is an unrealistic assumption in

practice, it is useful for illustrating methods and concepts discussed later in this work.

2.2 Maximum Likelihood Estimation

Technical issues for fitting GHMs receive considerable attention, as they present many difficulties. Maximum likelihood (ML) estimates of the parameter $\boldsymbol{\psi}$ are obtained using the *marginal* likelihood in which the unobservable, random effects are integrated out

$$L(\boldsymbol{\psi}|\mathbf{y}) \equiv \int f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi}) d\mathbf{u} = f(\mathbf{y}; \boldsymbol{\psi}). \tag{2.6}$$

In what follows we write $l(\boldsymbol{\psi}) \equiv \log L(\boldsymbol{\psi}|\mathbf{y})$ for the log of the likelihood function, suppressing the dependence on the data $\mathbf{y}$. The maximum likelihood estimator (MLE), $\hat{\boldsymbol{\psi}}$, is usually obtained as the solution to the scoring equations $\mathbf{S}(\boldsymbol{\psi}) = \mathbf{0}$, where the score function $\mathbf{S}$ is defined as the gradient of the log-likelihood, that is,

$$\mathbf{S}(\boldsymbol{\psi}) = \frac{\partial}{\partial \boldsymbol{\psi}} l(\boldsymbol{\psi}). \tag{2.7}$$

In GHMs the likelihood typically does not have a closed form expression. Except for a few special cases, the integral in (2.6) is analytically intractable. Consider the following example for illustration.

2.2.1 Logistic-Normal Model

The likelihood for the logistic-normal model in Section 2.1.1 is

$$L(\boldsymbol{\beta}, \sigma_1^2|\mathbf{y}) = \prod_{i=1}^{m} \left\{ \int \prod_{j=1}^{r} \frac{\exp(y_{ij}(\mathbf{x}'_{ij}\boldsymbol{\beta} + u_i))}{[1 + \exp(\mathbf{x}'_{ij}\boldsymbol{\beta} + u_i)]} \frac{\varphi(u_i/\sigma_1)}{\sigma_1} du_i \right\}, \tag{2.8}$$

where $\varphi(x)$ denotes the pdf of a standard normal random variable. The right hand side of equation (2.8) contains the product of m one-dimensional integrals. Notice, however, that none of these integrals has a closed form solution and therefore approximate methods must be used to evaluate the likelihood function.

An additional difficulty associated with GHMs is that the intractable integral in (2.6) often is of very high dimension. In the examples considered so far we have, for simplicity of exposition, assumed that the random effects u_i are univariate and uncorrelated. In practice, this is often not the case. Consider the following modification of the logistic-normal example from Section 2.1.1.

2.2.2 Logistic-Normal Model with Multivariate Random Effects

A simple modification of the logistic-normal model from Section 2.1.1 is achieved by supposing that data is available for T different years. Thus, we observe binary responses, y_{hij}, where y_{hij} is the performance of student j, $(j = 1, \ldots, r)$, in school i, $(i = 1, \ldots, m)$, from school-year h, $(h = 1, \ldots, T)$. We still want to account for an unobservable effect due to school i. However, the effect of school i may change over the course of T years. Let u_{hi} denote the effect from school i for year h, and let $\mathbf{u}_i = (u_{1i}, \ldots, u_{Ti})'$. We assume that $\mathbf{u}_i \sim N(\mathbf{0}, \mathbf{G}_i)$. This approach allows effects from different years (for the same school) to be correlated. The (modified) logistic-normal model is

$$\text{logit}(\pi_{hij}) = \mathbf{x}'_{hij}\boldsymbol{\beta} + u_{hi}. \tag{2.9}$$

Then, the likelihood function from equation (2.8) changes to

$$L(\boldsymbol{\beta}, \mathbf{G}_1, \ldots, \mathbf{G}_m|\mathbf{y}) = \prod_i \int \prod_{h,j} \frac{\exp(y_{hij}(\mathbf{x}'_{hij}\boldsymbol{\beta} + u_{hi}))}{[1 + \exp(\mathbf{x}'_{hij}\boldsymbol{\beta} + u_{hi})]} \frac{\exp\{-\frac{1}{2}\mathbf{u}'_i\mathbf{G}_i^{-1}\mathbf{u}_i\}}{|2\pi\mathbf{G}_i|^{-1/2}} d\mathbf{u}_i, \tag{2.10}$$

a product of m T-dimensional (intractable) integrals.

CHAPTER 3
DETERMINISTIC MAXIMUM LIKELIHOOD COMPUTATION

In this chapter we introduce several deterministic approaches for dealing with the intractable likelihood function in (2.6). Section 3.1 describes an analytical approach based on a Laplace approximation to the intractable integral. Section 3.2 introduces the general ideas of quadrature methods. The drawback to these first two methods is that they may not perform well in GHMs. Laplace approximations can produce inconsistent parameter estimates and quadrature methods are generally not recommended when the dimension of the integral is large.

In Sections 3.3 and 3.4 we proceed by introducing two iterative methods, the Newton-Raphson and the EM algorithm. We explain the general ideas of these two algorithms and illustrate them on several examples. In Section 3.5 we highlight the similarity between the two algorithms and discuss several important modifications of the EM algorithm in Section 3.6.

3.1 Penalized Quasi-Likelihood

One approach for dealing with the intractable integral in (2.6) is to use an analytical approximation. The penalized quasi-likelihood (PQL) methods of Breslow and Clayton (1993) and Wolfinger and O'Connell (1993) are essentially derived from a Laplace approximation to the integral in (2.6). The basic idea of the Laplace approximation (see, e.g. De Bruijn, 1958) is as follows. Suppose we want to approximate the integral

$$\int \exp\{ng(\mathbf{x})\}d\mathbf{x}, \tag{3.1}$$

which is the form of the GHM likelihood for distributions in the exponential family. Let $\hat{\mathbf{x}}$ be the mode of $g(\cdot)$, satisfying $\partial g(\mathbf{x})/\partial \mathbf{x}|_{\mathbf{x}=\hat{\mathbf{x}}} = \mathbf{0}$, and define $\boldsymbol{\Sigma} = -(\partial^2 g(\mathbf{x})/\partial \mathbf{x} \partial \mathbf{x}'|_{\mathbf{x}=\hat{\mathbf{x}}})^{-1}$. A second order Taylor expansion of the integrand in (3.1) about $\hat{\mathbf{x}}$ yields

$$\int \exp\{ng(\mathbf{x})\}d\mathbf{x} \approx \int \exp\left\{ng(\hat{\mathbf{x}}) - \frac{n}{2}(\mathbf{x}-\hat{\mathbf{x}})\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\hat{\mathbf{x}})'\right\} d\mathbf{x}. \tag{3.2}$$

But the rightmost term in equation (3.2) can be identified as the kernel of a normal random variable with mean $\hat{\mathbf{x}}$ and covariance matrix $\boldsymbol{\Sigma}/n$. It follows that

$$\int \exp\{ng(\mathbf{x})\}d\mathbf{x} \approx |2\pi\boldsymbol{\Sigma}/n|^{1/2} \exp\{ng(\hat{\mathbf{x}})\}. \tag{3.3}$$

Equation (3.3) is the basic form of the Laplace approximation.

However, PQL is now known to produce inconsistent parameter estimates and the approximation can be very poor when the random effects variance components are not small (Kuk, 1995; Breslow and Lin, 1995; Lin and Breslow, 1996; Jiang, 1998).

3.2 Quadrature

Another common approach for handling the intractable integral in (2.6) is numerical integration via Gauss-Hermite quadrature. Quadrature methods approximate integrals by sums based on weights and nodes computed from the integrand. The fundamental idea of Gauss-Hermite Quadrature is as follows. In order to approximate an integral of the form

$$\int f(x)e^{-x^2}dx, \tag{3.4}$$

which is the form of the GHM-likelihood in the case of normally distributed random effects, one has to find the roots $x_k, (k = 1, \ldots, K)$, of the Kth Hermite polynomial, H_K, and corresponding weights, w_k. These are tabulated in, for

example, Abramowitz and Stegun (1992). Then, one approximates (3.4) by

$$\int f(x)e^{-x^2}dx \approx \sum_{k=1}^{K} w_k f(x_k). \tag{3.5}$$

The weights are chosen so that, if $f(x)$ is a polynomial of maximal degree $(2K-1)$, the approximation in equation (3.5) is exact.

Adaptive Gauss-Hermite quadrature methods center the integrals about the mode of the integrand and decrease the number of nodes needed (Liu and Pierce, 1994). Adaptive Gauss-Hermite quadrature forms the basis for SAS's procedure NLMIXED (Version 7 and 8).

In principle, the error in (3.5) can be made arbitrarily small by increasing the number, K, of quadrature points. However, establishing exactly how the error (in the approximation of the integral) propagates through the maximization can be difficult (Crouch and Spiegelman, 1990). Moreover, the accuracy of quadrature methods can be very poor for higher dimensional integrals (Evans and Swartz, 1995).

3.3 The Newton-Raphson (NR) Algorithm

The Newton-Raphson (NR) algorithm is an iterative method that is often used to approximate the maximum (or minimum) of a function. Newton-Raphson is a popular tool in many fields and it is often used in statistics to approximate the MLE.

The heart of any iterative method is the rule it uses to find the next estimate given the current one. Essentially two decisions need to be made: In which direction will the next estimate be in relation to the current one; and how far will the next estimate be from the current one? These are termed the step direction and step size of the method. The direction of the fastest increase of the log likelihood is the vector of the first derivatives or gradient. Hence a general class of

gradient methods can be defined as follows: Given a current approximation, $\boldsymbol{\psi}^{(t)}$, to the MLE, determine the update $\boldsymbol{\psi}^{(t+1)}$ by

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} + s^{(t)}\mathbf{M}^{(t)}\mathbf{S}(\boldsymbol{\psi}^{(t)}) \tag{3.6}$$

In this equation $s^{(t)}$, a scalar, is the step size for the tth step, $\mathbf{M}^{(t)}$ is a modifier matrix for the tth step and $\mathbf{S}(\boldsymbol{\psi})$ is the score function defined in (2.7). The method of steepest ascent is a special case of (3.6) with $\mathbf{M}^{(t)} = \mathbf{I}$. Unfortunately this particular choice performs very poorly in practice (Bard, 1974), tending to converge very slowly.

The Newton-Raphson method on the other hand replaces $s^{(t)}\mathbf{M}^{(t)}$ in (3.6) by $-\mathbf{H}(\boldsymbol{\psi}^{(t)})^{-1}$, where

$$\mathbf{H}(\boldsymbol{\psi}) = \frac{\partial^2}{\partial\boldsymbol{\psi}\partial\boldsymbol{\psi}'}l(\boldsymbol{\psi}) = \frac{\partial}{\partial\boldsymbol{\psi}'}\mathbf{S}(\boldsymbol{\psi}) \tag{3.7}$$

is the Hessian matrix of the log-likelihood [1] . Thus, the $(t+1)$st NR update has the form

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} - \mathbf{H}(\boldsymbol{\psi}^{(t)})^{-1}\mathbf{S}(\boldsymbol{\psi}^{(t)}). \tag{3.8}$$

Newton-Raphson enjoys local quadratic convergence to a solution of the scoring equations (e.g. Bronstein and Semendjajew, 1991). Convergence at a quadratic rate means that the number of correct decimals approximately doubles in each iteration. More precisely, if $\boldsymbol{\psi}^{(t)}$ and $\boldsymbol{\psi}^{(t+1)}$ are two successive approximations to the MLE, $\hat{\boldsymbol{\psi}}$, then

$$|\boldsymbol{\psi}^{(t+1)} - \hat{\boldsymbol{\psi}}| \approx b|\boldsymbol{\psi}^{(t)} - \hat{\boldsymbol{\psi}}|^2, \tag{3.9}$$

[1] Within the context of Newton-Raphson, it is common to refer to $\mathbf{H}(\boldsymbol{\psi})$ as the Hessian matrix. Notice, however, that $-\mathbf{H}(\hat{\boldsymbol{\psi}})$ equals the observed information matrix, which is often denoted by $\mathbf{J}_o$ (see Section 7.1).

where $b > 0$. We emphasize, however, that NR's quadratic convergence rate typically only holds *locally*, in a neighborhood of the MLE. Therefore, equation (3.9) may only be true close to convergence of the algorithm.

The popularity of NR as an optimization routine is based on the fact that most other methods do *not* achieve a quadratic convergence rate. However, NR's fast rate of convergence is often offset by several disadvantages. One disadvantage is that NR requires the computation of the gradient and the Hessian of the log-likelihood at each iteration which can be a very tedious computational task. Moreover, at each iteration the inverse of the Hessian has to be calculated. This is generally done numerically which can lead to severe inaccuracies due to rounding error (see Searle et al., 1992, p.142). Several modifications of NR avoid these problems. Quasi-Newton techniques, for example, replace the inverse of the Hessian by an approximation that only uses the gradient of the log-likelihood. The downside of these modifications is that they do not retain NR's quadratic rate of convergence.

Another disadvantage of NR is that it only converges for "sufficiently good" starting values in the neighborhood of the MLE. Without extra computational effort to detect good starting values, NR can easily diverge to values on the boundary of the parameter space and produce meaningless parameter estimates, such as negative variance estimates (Thompson and Meyer, 1986; Callanan and Harville, 1991). But choosing "good" starting values requires some prior knowledge about the solution, which in practice is often not available. Alternatively, "trial-and-error" methods or "grid-search" procedures can be used to find appropriate starting points. However, these methods can prove to be very time consuming, especially in high dimensional problems.

We now consider two examples to illustrate and further investigate the performance of the Newton-Raphson algorithm.

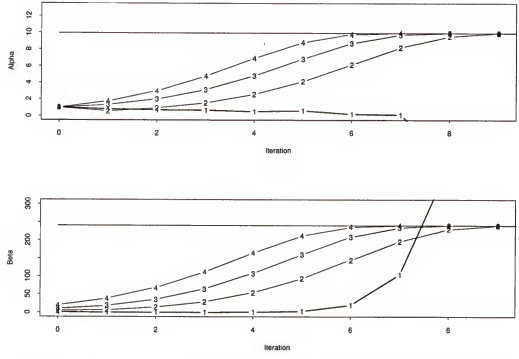

Figure 3.1: Convergence of Newton-Raphson in the beta-binomial model.

Example: OWMM. Consider the OWMM introduced in Section 2.1.3. We show in the Appendix (Section A.4, equations (A.18) and (A.19)), that the score function and the Hessian are

$$S(\mu) = -\frac{mr}{\sigma_0^2}(1-b)(\mu-\hat{\mu}),\ H(\mu) = -\frac{mr}{\sigma_0^2}(1-b), \tag{3.10}$$

where $b = (r\sigma_1^2)/(r\sigma_1^2+\sigma_0^2)$ and $\hat{\mu} = \bar{y}_{..}$. It follows that the $(t+1)$st NR update is

$$\mu^{(t+1)} = \mu^{(t)} - H(\mu^{(t)})^{-1}S(\mu^{(t)}) = \hat{\mu}. \tag{3.11}$$

That is, NR converges in one iteration. This was to be expected as for this model the log-likelihood is a quadratic function in μ and NR, by construction, finds the extreme value of a quadratic function in only one iteration (see, e.g., Searle et al., 1992).

Example: Beta-Binomial Model. Consider the beta-binomial model introduced in Section 2.1.1. Notice that the (marginal) likelihood is available in closed form. In particular,

$$L(\alpha, \beta; \mathbf{y}) = \prod_{i=1}^{13} \binom{n_i}{y_i} \frac{B(y_i + \alpha; n_i - y_i + \beta)}{B(\alpha; \beta)}, \tag{3.12}$$

where $B(x; y)$ denotes the beta function with parameters x and y. For the data in Table 2.1, Booth and Caffo (2001) report that the MLE is $(\hat{\alpha}; \hat{\beta}) = (9.95; 240.8)$.

Figure 3.1 shows the convergence behavior of NR for maximizing the likelihood function in equation (3.12) for several different starting values. Lines 1 through 4 are the iteration histories for the starting values $(\alpha^{(0)}; \beta^{(0)}) \in \{(1, 2), (1, 5), (1, 10), (1, 20)\}$, respectively. Notice that the starting value for the α-component is constant over all four runs. Also, compared to the target value, $\hat{\beta} = 240.8$, the changes in β can be considered rather moderate.

Notice that the behavior of NR is similar for the values $\beta^{(0)} = 5, 10$ and 20, showing some improvement as $\beta^{(0)}$ increases. However, NR fails for $\beta^{(0)} = 2$; for this value, the method breaks down and diverges to the boundary of the parameter space. This indicates that even small changes in the starting value can have a huge impact on the performance of Newton-Raphson.

3.4 The EM Algorithm

Another method that is often used to find the MLE is the EM algorithm. The earliest application of EM dates back to McKendrick (1926). However, it was not until the work of Dempster et al. (1977) that the full power of the algorithm began to be appreciated.

The EM algorithm is an iterative procedure, in which each iteration consists of the following two steps. In the E-Step one computes the conditional expectation of the complete data log-likelihood, conditional on the observed data and the current

parameter estimate $\boldsymbol{\psi}^{(t)}$,

$$Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t)}) = E\left(\log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi}) \middle| \mathbf{y}; \boldsymbol{\psi}^{(t)}\right). \tag{3.13}$$

In the M-Step the parameter estimate is updated by the maximizer of (3.13), that is, by the value $\boldsymbol{\psi}^{(t+1)}$ that satisfies

$$Q(\boldsymbol{\psi}^{(t+1)}|\boldsymbol{\psi}^{(t)}) \geq Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t)}) \tag{3.14}$$

for all values of $\boldsymbol{\psi}$ in the parameter space. Assuming that the derivative can be passed through the expectation in (3.13), the maximization is generally accomplished by solving the EM estimating equations

$$\mathbf{F}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}) \equiv E\left(\frac{\partial}{\partial \boldsymbol{\psi}} \log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi}) \middle| \mathbf{y}; \boldsymbol{\psi}^{(t)}\right) = \mathbf{0}. \tag{3.15}$$

Equation (3.15) defines the parameter update as an implicit function of $\boldsymbol{\psi}^{(t)}$;

$$\boldsymbol{\psi}^{(t+1)} = \mathbf{M}(\boldsymbol{\psi}^{(t)}) \tag{3.16}$$

say. The MLE is a fixed point of the function $\mathbf{M}$; that is $\hat{\boldsymbol{\psi}} = \mathbf{M}(\hat{\boldsymbol{\psi}})$. Expanding $\mathbf{M}(\boldsymbol{\psi}^{(t)})$ in a first order Taylor series about the fixed point results in the approximate relation

$$\boldsymbol{\psi}^{(t+1)} - \hat{\boldsymbol{\psi}} \approx \mathbf{B}\,[\boldsymbol{\psi}^{(t)} - \hat{\boldsymbol{\psi}}], \tag{3.17}$$

where $\mathbf{B}(\boldsymbol{\psi}) = \partial \mathbf{M}(\boldsymbol{\psi})/\partial \boldsymbol{\psi}'$ and

$$\mathbf{B} \equiv \mathbf{B}(\hat{\boldsymbol{\psi}}). \tag{3.18}$$

The matrix $\mathbf{B}$ is often called the *rate matrix* (e.g. Meng and Rubin, 1991) as it governs the rate of convergence of the algorithm. Notice that the smaller the elements of $\mathbf{B}$, the faster EM converges to the MLE. To be consistent with the common notion that larger values indicate faster convergence, some authors define the *speed matrix* as $\mathbf{I} - \mathbf{B}$ (see Meng, 1994).

EM is often said to converge at a linear rate. This is presumably because equation (3.17) shows that the error after $(t+1)$ iterations is linearly related to the error after t iterations. Thus, EM's convergence is slower than the quadratic rate of Newton-Raphson.

The reasons for EM's popularity, compared with other numerical methods, are its superior stability properties. Dempster et al. (1977) and Wu (1983) show that the algorithm increases the value of the observed log-likelihood at each iteration. Redner and Walker (1984, p.201) note that "EM [...] has been found in most instances to have the advantage of a reliable global convergence, low cost per iteration, economy of storage and ease of programming, as well as certain heuristic appeal."

However, it is EM's rate of convergence that is often criticized. Indeed, Redner and Walker (1984, p.201) also comment that "EM's convergence can be maddeningly slow in simple problems which are often encountered in practice." Rubin (1991) points out that the speed of convergence of EM is proportional to $1 - b$, where b is the fraction of missing information. The fraction of missing information is defined as the ratio of missing-to-complete information, or equivalently, as one minus the ratio of observed-to-complete information (see Section 7.1 for a more thorough discussion of this issue). Consider the following example for illustration.

Example: OWMM. We show in the Appendix (Section A.4, equation (A.17)), that the $(t+1)$st EM update for the OWMM is

$$\mu^{(t+1)} = \hat{\mu} + b(\mu^{(t)} - \hat{\mu}), \tag{3.19}$$

where $b = \sigma_1^2/(\sigma_1^2 + \sigma_0^2/r)$. Notice that $0 \leq b < 1$. Equation (3.19) implies that the approximate relation in (3.17) holds exactly. Furthermore, the error, $\mu^{(t+1)} - \hat{\mu}$, of the EM update after $(t+1)$ iterations equals $b^{t+1}d_0$, where $d_0 = \mu^{(0)} - \hat{\mu}$, the initial error. Therefore, the smaller the value of b, the faster EM will converge to

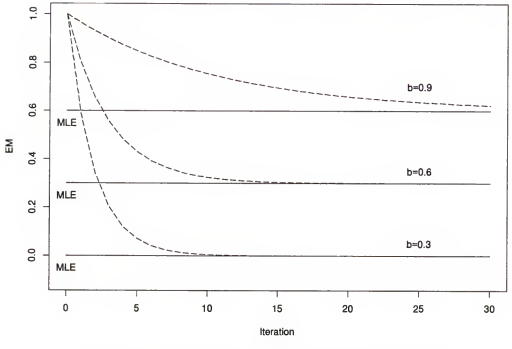

Figure 3.2: EM's rate of convergence in the OWMM.

the MLE. But notice that b measures the fraction of missing information (about μ). Indeed, notice that the observed data information is

$$-\left.\frac{\partial^2 \log L(\mu|\mathbf{y})}{\partial \mu^2}\right|_{\mu=\hat{\mu}} = \frac{m}{\sigma_1^2 + \sigma_0^2/r}. \tag{3.20}$$

Similarly, the complete data information is (see Section 7.1 for more details)

$$-E\left[\left.\frac{\partial^2 \log f(\mathbf{y}, \mathbf{u}; \mu)}{\partial \mu^2}\right| \mathbf{y}; \mu\right]\Bigg|_{\mu=\hat{\mu}} = \frac{m}{\sigma_0^2/r}. \tag{3.21}$$

Thus, one minus the ratio of observed-to-complete information is

$$1 - \frac{\sigma_0^2/r}{\sigma_1^2 + \sigma_0^2/r} = b;$$

that is, the fraction of missing information.

Figure 3.2 displays EM's rate of convergence for increasing fractions of missing information, $b \in \{0.3, 0.6, 0.9\}$. For each of these three values of b a set of data was

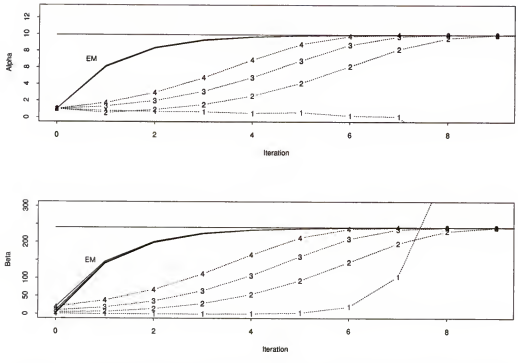

Figure 3.3: Convergence of the EM algorithm in the beta-binomial model.

simulated from the OWMM (with fixed parameters $m = 10; r = 10; \mu = 0; \sigma_0^2 = 1$), resulting in the MLEs, $0, 0.3$ and 0.6, respectively. These are displayed by the horizontal lines in Figure 3.2. Next, the EM algorithm was applied to each of the three data sets with the same starting value, $\mu^{(0)} = 1$. The dashed lines show the histories of the first 30 EM-iterations.

Notice that for $b = 0.3$, the MLE is the furthest away from the starting value. However, the algorithm converges at the fastest pace, reaching the MLE after about 10 iterations. Conversely, for $b = 0.9$, the MLE is closest to the starting value. But even after 30 iterations, EM has not converged yet. This illustrates that the smaller the fraction of missing information, the faster is EM's rate of convergence.

The following example allows us to compare the convergence behavior of EM with that of Newton-Raphson. In addition, it also sheds light on the stability of both algorithms with respect to the choice of the starting value.

Example: Beta-Binomial Model. Consider again the beta-binomial model introduced in Section 2.1.1. In Figure 3.1 we illustrated the convergence of NR for this model using four different starting values. Figure 3.3 shows the convergence of EM for the same four starting values. In particular, the iteration histories of EM are given by the four solid lines in Figure 3.3, whereas the dashed lines reproduce the corresponding iteration histories of NR from Figure 3.1.

Our first observation is that the four runs of EM are almost indistinguishable from each other. (Especially for the α-component, all four iteration histories seem to lie on the same line). This implies that EM's convergence is almost identical for each of the four starting values. In contrast, recall that NR's behavior strongly depended on the starting point. This observation certainly suggests that EM is a more stable algorithm; moderate changes in the starting values barely affect the convergence of EM. In particular, recall that NR did not converge at all for the first starting value $(\alpha^{(0)}; \beta^{(0)}) = (1; 2)$. However, this point does not cause any problems for EM at all.

A second observation in Figure 3.3 is that EM reaches the *neighborhood* of the MLE faster than NR. Clearly, at the first two or three iterations, EM takes bigger steps than NR. However, it is after iteration three that EM's criticized slow convergence sets in. It can be seen that the relative gain between two successive updates after iteration three is much smaller for EM than it is for NR; close to the MLE NR clearly converges at a faster rate.

3.5 Analytical Comparison of Newton-Raphson and EM

Recall that the EM update solves the equation $\mathbf{F}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}) = \mathbf{0}$ in (3.15). In most practical circumstances, however, there will be no explicit solution to this equation and an iterative routine, such as Newton-Raphson, is needed to find the EM update $\boldsymbol{\psi}^{(t+1)}$. That is, $\boldsymbol{\psi}^{(t+1)}$ is the limit of the sequence $\{\boldsymbol{\psi}^{(t,r)}\}_{r=0}^{\infty}$, where

$\boldsymbol{\psi}^{(t,0)} = \boldsymbol{\psi}^{(t)}$ and for $r \geq 0$,

$$\boldsymbol{\psi}^{(t,r+1)} = \boldsymbol{\psi}^{(t,r)} - \left(\frac{\partial \mathbf{F}}{\partial \boldsymbol{\psi}'}\right)^{-1}\Bigg|_{\boldsymbol{\psi}=\boldsymbol{\psi}^{(t,r)}} \mathbf{F}(\boldsymbol{\psi}^{(t,r)}, \boldsymbol{\psi}^{(t)}). \tag{3.22}$$

Typically, the $(t+1)$st step of EM consists of iterating (3.22) until convergence. However, a modification of the EM algorithm, proposed by Lange (1995), is to complete only *one* NR iteration of (3.22) in the $(t+1)$st step. Specifically, if we define

$$\mathbf{H}_Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}) \equiv \frac{\partial}{\partial \boldsymbol{\psi}'}\mathbf{F}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}) = \frac{\partial^2}{\partial \boldsymbol{\psi}\partial \boldsymbol{\psi}'}Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t)}), \tag{3.23}$$

the Hessian of the Q-function in (3.13) [2] . Let $\mathbf{H}_Q(\boldsymbol{\psi}^{(t)}) \equiv \mathbf{H}_Q(\boldsymbol{\psi}^{(t)}, \boldsymbol{\psi}^{(t)})$. Lange (1995) defines a modified EM algorithm as

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} - \mathbf{H}_Q(\boldsymbol{\psi}^{(t)})^{-1}\mathbf{F}(\boldsymbol{\psi}^{(t)}, \boldsymbol{\psi}^{(t)}). \tag{3.24}$$

On the other hand, recall that Newton-Raphson in (3.8) was defined as

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} - \mathbf{H}(\boldsymbol{\psi}^{(t)})^{-1}\mathbf{S}(\boldsymbol{\psi}^{(t)}), \tag{3.25}$$

where $\mathbf{S}$ and $\mathbf{H}$ are the gradient and the Hessian of the log-likelihood, respectively. A close relationship between algorithms (3.24) and (3.25) is now revealed by the identity

$$\mathbf{S}(\boldsymbol{\psi}) = E\left[\frac{\partial}{\partial \boldsymbol{\psi}}\log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi})\Big| \mathbf{y}; \boldsymbol{\psi}\right] \tag{3.26}$$

(refer to the Appendix, Section A.1, for details). This implies that $\mathbf{S}(\boldsymbol{\psi}) \equiv \mathbf{F}(\boldsymbol{\psi}, \boldsymbol{\psi})$ and thus the difference between the algorithms (3.24) and (3.25) is only due to the difference between the two Hessians, $\mathbf{H}_Q = \partial \mathbf{F}/\partial \boldsymbol{\psi}'$ and $\mathbf{H} = \partial \mathbf{S}/\partial \boldsymbol{\psi}'$.

[2] Similar to the negative Hessian, $-\mathbf{H}(\hat{\boldsymbol{\psi}})$, which is often referred to as the *observed information*, $-\mathbf{H}_Q(\hat{\boldsymbol{\psi}})$ is also called the *complete information* and denoted by $\mathbf{J}_c$ (see Section 7.1).

Furthermore, we show in the Appendix (Section A.2) that

$$\mathbf{H}(\boldsymbol{\psi}) = \mathbf{H}_Q(\boldsymbol{\psi}) + \mathrm{Var}\left(\frac{\partial}{\partial \boldsymbol{\psi}} \log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi}) \middle| \mathbf{y}; \boldsymbol{\psi}\right), \tag{3.27}$$

where

$$\mathrm{Var}\left(\frac{\partial}{\partial \boldsymbol{\psi}} \log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi}) \middle| \mathbf{y}; \boldsymbol{\psi}\right)\Big|_{\boldsymbol{\psi}=\hat{\boldsymbol{\psi}}} \tag{3.28}$$

is often referred to as the missing information. (See Section 7.1 for a more thorough discussion of this issue.)

However, since $\mathrm{Var}(\partial \log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi})/\partial \boldsymbol{\psi} | \mathbf{y}; \boldsymbol{\psi})$ is semi-positive definite, it follows from equation (3.27) that $\mathbf{H}(\boldsymbol{\psi}) \leq \mathbf{H}_Q(\boldsymbol{\psi})$ [3] . This means that the NR step sizes are larger than those of the modified EM algorithm in (3.24), which certainly helps to explain why typically EM is more stable than NR.

We want to emphasize that this last paragraph in no way contradicts our previous findings. Recall that in Figure 3.3 we observed that at the initial iterations (between iteration number one and three, say) EM takes larger steps than NR. However, also recall that in this example we used the *original* EM algorithm, which executes a complete M-step in each iteration (in contrast to Lange's modified EM in (3.24), which only carries out one NR-iteration of the M-step). To compare the performance of the original EM with Lange's modified version, refer to Figure 3.4. It shows the first nine iterations of EM (solid lines), NR (dashed lines, labeled "1") and Lange's EM (dashed lines, labeled "2"). Clearly, the convergence behavior of Lange's modified EM resembles more NR than it does EM. Notice in particular the smaller step sizes of modified EM (compared to NR) in the later iterations.

[3] For two symmetric matrices, $\mathbf{A}$ and $\mathbf{B}$, we define $\mathbf{A} \leq \mathbf{B}$, if $\mathbf{x}'(\mathbf{A}-\mathbf{B})\mathbf{x} \leq 0, \forall \mathbf{x}$.

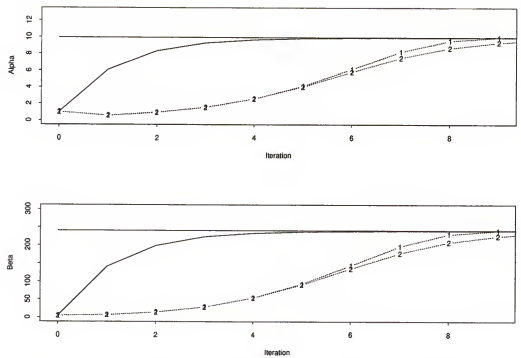

Figure 3.4: EM, Lange's modified EM and NR in the beta-binomial model.

3.6 Modifications of EM

In this section we discuss several important modifications of the EM algorithm. We start with a modification that facilitates the computation of the M-step.

Meng and Rubin (1993) introduce the ECM (Expectation / Conditional Maximization) algorithm for situations where the EM algorithm is unattractive because the M-step is computationally difficult. The ECM algorithm takes advantage of the simplicity of complete-data conditional maximum likelihood estimation by replacing a complicated M-step of EM with several computationally simpler CM-steps. Meng and Rubin show that the ECM algorithm shares the same convergence properties of EM, such as increasing the likelihood at every iteration. Their algorithm takes the following form. Let the parameter of interest, $\boldsymbol{\psi}$, be partitioned into $\mathcal{S}$ (possibly vector-valued) components, $\boldsymbol{\psi} = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{\mathcal{S}})$. For $s = 1, \ldots, \mathcal{S}$, let $\boldsymbol{\psi}^{(t+\frac{s}{\mathcal{S}})} \equiv (\boldsymbol{\psi}_1^{(t+1)}, \boldsymbol{\psi}_2^{(t+1)}, \ldots, \boldsymbol{\psi}_s^{(t+1)}, \boldsymbol{\psi}_{s+1}^{(t)}, \ldots, \boldsymbol{\psi}_{\mathcal{S}}^{(t)})$ be the $s/\mathcal{S}$-fraction of the full

$(t+1)$st update; that is, the parameter estimate obtained after the first s conditional M-steps. Then in the $(s+1)$st CM-step of the $(t+1)$st iteration, ECM finds the value $\boldsymbol{\psi}^{(t+\frac{s+1}{S})}$ such that

$$Q(\boldsymbol{\psi}^{(t+\frac{s+1}{S})}|\boldsymbol{\psi}^{(t+\frac{s}{S})}) \geq Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t+\frac{s}{S})}),\ \forall \boldsymbol{\psi} \in \Psi_{s+1},$$

where Ψ denotes the parameter space and Ψ_{s+1} the subspace that satisfies $\Psi_{s+1} = \{\boldsymbol{\psi} \in \Psi : \boldsymbol{\psi} = (\boldsymbol{\psi}_1^{(t+1)}, \boldsymbol{\psi}_2^{(t+1)}, \ldots, \boldsymbol{\psi}_s^{(t+1)}, \boldsymbol{\psi}_{s+1}, \boldsymbol{\psi}_{s+2}^{(t)}, \ldots, \boldsymbol{\psi}_S^{(t)})\}$.

Liu and Rubin (1994) build on the above idea and develop the ECME (Expectation/ Conditional Maximization Either) algorithm, which is similar to the ECM algorithm, but where some of the CM-steps of ECM, which maximize the constrained expected complete-data log likelihood function, are replaced by maximizing the constrained actual likelihood function. The authors show that this algorithm shares the same stable monotone convergence as EM and ECM, but can be substantially faster than either EM or ECM, measured using actual computing time. The drawback is that in most cases where the EM algorithm is useful, the actual likelihood function is not available in closed form and hence this method will not be applicable in many situations of interest.

The two previous methods are in principle not designed to speed up the convergence of the EM algorithm. Their motivation is to simplify the M-step and save computing time. However, they generally do not decrease the number of iterations needed. Several methods have been proposed under the title "EM accelerators". One of these methods is Aitken acceleration investigated by Laird et al. (1987) and Louis (1982).

Laird et al. (1987) discuss a univariate and a multivariate Aitken accelerator. We consider the univariate case first. There, if b denotes the univariate analogue of the rate matrix $\mathbf{B}$ then it follows from equation (3.17) that, if b is known, after the

$(t+1)$st iteration the MLE can be *predicted* via

$$\hat{\psi} \approx \psi^{(t)} + \frac{1}{1-b}(\psi^{(t+1)} - \psi^{(t)}). \tag{3.29}$$

The Aitken accelerated EM algorithm uses the predicted MLE in place of the true EM update; that is, the $(t+1)$st EM update is replaced by the right hand side of equation (3.29).

In the multivariate case, $1/(1-b)$ is replaced by $(\mathbf{I}-\mathbf{B})^{-1}$; specifically, in multivariate Aitken acceleration the MLE is predicted according to

$$\hat{\boldsymbol{\psi}} \approx \boldsymbol{\psi}^{(t)} + (\mathbf{I}-\mathbf{B})^{-1}(\boldsymbol{\psi}^{(t+1)} - \boldsymbol{\psi}^{(t)}). \tag{3.30}$$

In practice, b and $\mathbf{B}$ in equations (3.29) and (3.30) have to be estimated. Laird et al. propose to estimate b by

$$\hat{b} = \frac{\psi^{(t+1)} - \psi^{(t)}}{\psi^{(t)} - \psi^{(t-1)}}. \tag{3.31}$$

A multivariate extension of this idea leads to an estimate for $\mathbf{B}$.

Example: Beta-Binomial Model. Figure 3.5 illustrates the performance of the Aitken accelerated EM algorithm in the beta-binomial model. For the starting value $(\alpha^{(0)};\beta^{(0)}) = (1;2)$, Figure 3.5 shows the first 5 iterations of EM (dashed line) and accelerated EM (solid line). We started to predict the MLE after the first iteration; that is, for iterations $t = 2,\ldots,5$, we replaced the true EM update by the predicted MLE in equation (3.30).

Notice that the accelerated EM algorithm performs very well in the beta-binomial model. The first prediction step (given by iteration 2 in Figure 3.5) brings the accelerated EM algorithm into a close neighborhood of the MLE, whereas the EM algorithm is still far from convergence at this point.

Acceleration methods do not always work well, however. Laird et al. suggest checking that the prediction actually increases the likelihood over the true EM

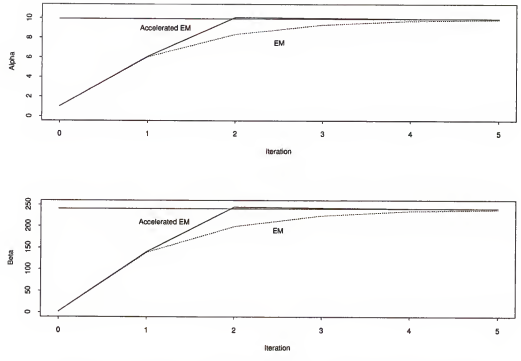

Figure 3.5: Aitken accelerated EM in the beta-binomial model.

update. Moreover, equations (3.29) and (3.30) only provide good approximations in a neighborhood of the MLE; therefore the prediction will only improve over the true EM update when the algorithm is close to convergence.

Many other modifications exist under the name "EM accelerators". These include Newton-Raphson type accelerators by Meilijson (1989) and Jamshidian and Jennrich (1993), and an acceleration based on an expansion of the parameter vector by Liu et al. (1998). However, these modifications are beyond the scope of this dissertation.

CHAPTER 4
STOCHASTIC MAXIMUM LIKELIHOOD COMPUTATION

4.1 Introduction

In Chapter 3 we introduced Newton-Raphson and the EM algorithm, two popular methods to find the maximum of the likelihood function. However, in GHMs neither of these methods are typically not directly applicable.

Indeed, we have argued that the log of the likelihood function in (2.6) is typically analytically intractable for GHMs. Thus its derivative, the score function $\mathbf{S}$, which is needed in every iteration of NR, is generally not available in closed form. Similarly, for the EM algorithm we need to compute the Q-function in (3.13), which is defined as the conditional expectation with respect to the density $g(\mathbf{u}|\mathbf{y};\boldsymbol{\psi})$, where $g(\mathbf{u}|\mathbf{y};\boldsymbol{\psi}) \propto f(\mathbf{y}|\mathbf{u};\boldsymbol{\psi}_1)g(\mathbf{u};\boldsymbol{\psi}_2)$. But since the normalizing constant for this density is the likelihood function in (2.6), Q (and its derivative $\mathbf{F}$) are typically not available in closed form.

Therefore, modifications of Newton-Raphson and the EM algorithm are needed in order to make these algorithms suitable for GHMs. In particular, appropriate approximations to the intractable integrals in $\mathbf{S}$ and $\mathbf{F}$ need to be found. In principle, these integrals can be approximated using deterministic methods (for example, using a Laplace approximation or numerical integration). However, a disadvantage of deterministic approaches is that the error of approximation is typically difficult to estimate. Booth and Hobert (1999) use this point to motivate the use of *Monte Carlo integration*.

The fundamental idea of Monte Carlo integration is as follows (see, e.g., Robert and Casella, 1999): Suppose we want to approximate an integral of the

form

$$I = \int f(u)g(u)du, \tag{4.1}$$

where g is a probability density. If we can draw a random (independent and identical distributed) sample $u^{(1)}, \ldots, u^{(M)}$ from the density g, then we can approximate (4.1) by the empirical average

$$\tilde{I} = \frac{1}{M}\sum_{k=1}^{M} f(u^{(k)}). \tag{4.2}$$

Notice that $\tilde{I}$ is an unbiased estimate for I and, by the Strong Law of Large Numbers, $\tilde{I}$ converges almost surely to I (as $M \to \infty$).

In contrast to deterministic approaches described in Chapter 3, the approximation in equation (4.2) is of *stochastic* nature; that is, the error of approximation (the *Monte Carlo error*), $\tilde{I} - I$, is a random variable that has mean zero and variance

$$\mathrm{Var}(\tilde{I}) = \frac{1}{M}\int (f(u) - I)^2 g(u)du. \tag{4.3}$$

If $\int f^2(u)g(u)du < \infty$, then $\mathrm{Var}(\tilde{I})$ can be estimated from the sample $u^{(1)}, \ldots, u^{(M)}$ through

$$\tilde{s}^2 = \frac{1}{M^2}\sum_{k=1}^{M}(f(u^{(k)}) - \tilde{I})^2. \tag{4.4}$$

Then by the Central Limit Theorem and Slutzky's Theorem,

$$(\tilde{I} - I)/\tilde{s} \xrightarrow{d} \mathcal{N}(0,1), \text{ as } M \to \infty. \tag{4.5}$$

Thus, equation (4.5) can be used to construct confidence bounds on $\tilde{I} - I$.

In this chapter we focus on *stochastic optimization* (or *Monte Carlo optimization*). Monte Carlo integration and Monte Carlo optimization are often closely related. For example, combining the ideas of Monte Carlo integration with the concepts of a deterministic optimization technique, such as the EM algorithm, leads to a stochastic optimization method.

We shall describe several stochastic optimization methods in this chapter. In Section 4.2 we describe the method of simulated maximum likelihood. In Section 4.3 we introduce the Monte Carlo EM and the Monte Carlo Newton-Raphson algorithms, stochastic versions of EM and NR. We then describe the fundamental ideas behind the stochastic approximation method introduced by Robbins and Monro (1951). Combining the concepts of stochastic approximation with those of EM and NR leads to two further algorithms, the stochastic approximation EM and the stochastic approximation Newton-Raphson algorithms. These two algorithms are introduced in Section 4.4.

4.2 Simulated Maximum Likelihood (SML)

Consider the likelihood function (2.6). The method of simulated maximum likelihood approximates the intractable integral in (2.6) using Monte Carlo integration. Specifically, let $h(\mathbf{u})$ denote an importance sampling distribution whose support includes that of $g(\cdot;\boldsymbol{\psi}_2)$. For example, one might choose $h(\mathbf{u}) = g(\mathbf{u};\boldsymbol{\psi}_2^*)$, where $\boldsymbol{\psi}_2^*$ is an initial guess at $\boldsymbol{\psi}_2$. An equivalent representation of the likelihood in (2.6) is

$$L(\boldsymbol{\psi}|\mathbf{y}) = \int f(\mathbf{y}|\mathbf{u};\boldsymbol{\psi}_1)\frac{g(\mathbf{u};\boldsymbol{\psi}_2)}{h(\mathbf{u})}h(\mathbf{u})d\mathbf{u}. \tag{4.6}$$

Equation (4.6) suggests a Monte Carlo approximation to the likelihood function of the form

$$\tilde{L}(\boldsymbol{\psi}|\mathbf{y}) = \frac{1}{M}\sum_{k=1}^{M} f(\mathbf{y}|\mathbf{u}^{(k)};\boldsymbol{\psi}_1)\frac{g(\mathbf{u}^{(k)};\boldsymbol{\psi}_2)}{h(\mathbf{u}^{(k)})}, \tag{4.7}$$

where $\mathbf{u}^{(1)}, ..., \mathbf{u}^{(M)}$ is a random sample from h. This gives an unbiased estimate of the likelihood regardless of the choice of h. The SML estimator, $\tilde{\boldsymbol{\psi}}$, maximizes (4.7). That is, $\tilde{\boldsymbol{\psi}}$ satisfies

$$\tilde{L}(\tilde{\boldsymbol{\psi}}|\mathbf{y}) \geq \tilde{L}(\boldsymbol{\psi}|\mathbf{y}), \forall \boldsymbol{\psi}.$$

One problem associated with SML is to obtain an initial guess at $\boldsymbol{\psi}_2$. A solution is to use a two- or multi-stage implementation of SML, where the current importance

sampling distribution is chosen based on the maximizer, ψ_2^*, from the previous stage (see, e.g., Geyer and Thompson, 1992; McCulloch, 1997).

In the artificial case, when ψ_2 is known and one can simulate directly from the random effects distribution g, equation (4.7) simplifies to

$$\tilde{L}(\psi|\mathbf{y}) = \frac{1}{M}\sum_{k=1}^{M} f(\mathbf{y}|\mathbf{u}^{(k)};\psi_1), \tag{4.8}$$

where $\mathbf{u}^{(1)},...,\mathbf{u}^{(M)}$ is a random sample from $g(\mathbf{u};\psi_2)$.

Example: OWMM. Consider the OWMM from Section 2.1.3. If we assume that the variance components, σ_0^2 and σ_1^2, are known and that we are only interested in estimating μ, then we can use approximation (4.8) with importance sampling distribution $N(0,\sigma_1^2)$. Figure 4.1 shows the plots of 20 simulated likelihood functions based on (4.8) for different Monte Carlo sample sizes, $M \in \{5, 50\}$, for a simulated set of data from the OWMM (with parameters $m = r = 2$, $\sigma_0^2 = \sigma_1^2 = 1$ and MLE $\hat{\mu} = 0$). Due to the introduction of Monte Carlo error, the simulated likelihood functions vary randomly in shape as well as in location about the MLE. Notice, however, how the variability decreases with increasing M.

Importance Sampling for SML. The main practical problem when applying SML is choosing an appropriate importance sampling distribution h. In principle, almost every choice of h will produce an unbiased and consistent estimator of the likelihood function (as long as the support of h includes that of g). However, some choices are better than others. Indeed, some importance sampling distributions can result in infinite variances for $\tilde{L}$ (see, e.g., Robert and Casella, 1999). In particular, importance sampling distributions h with *lighter tails* than g (that is, those with unbounded ratios g/h) are *not* recommended.

Many methods for constructing good importance sampling distributions have been suggested in the literature. These include using a mixture of several importance distributions and stratified sampling; using control and antithetic

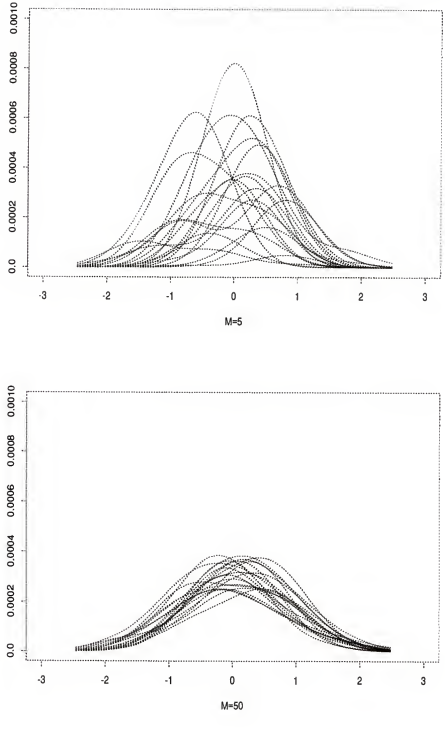

Figure 4.1: 20 simulated likelihood functions for the OWMM for different M.

variables; and also the use of non-parametric methods (see, e.g., Oh and Berger, 1993; Zhang, 1996; Owen and Zhou, 2000). However, the choice of the appropriate method often depends on the model and the data, making the search for good importance sampling distributions more of an art than a science.

4.3 Monte Carlo EM and Monte Carlo Newton-Raphson

4.3.1 Monte Carlo EM (MCEM)

Recall that the EM algorithm described in Section (3.4) requires the evaluation of the Q-function in equation (3.13) or its derivative, $\mathbf{F}$, in (3.15). However, we have argued at the beginning of this chapter that Q and $\mathbf{F}$ are typically analytically intractable in GHMs. In this case, a Monte Carlo version of the EM algorithm often offers an appropriate alternative.

Wei and Tanner (1990) propose to approximate an intractable Q-function by Monte Carlo integration. In particular, let $\boldsymbol{\psi}^{(t)}$ be the current MCEM update and let $\mathbf{u}^{(t,1)}, ..., \mathbf{u}^{(t,M_t)}$ be a random sample of size M_t from $g(\mathbf{u}|\mathbf{y}; \boldsymbol{\psi}^{(t)})$, the conditional distribution of $\mathbf{u}$ given $\mathbf{y}$ evaluated at $\boldsymbol{\psi} = \boldsymbol{\psi}^{(t)}$. Then, a Monte Carlo approximation to Q is

$$\tilde{Q}(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t)}) = \frac{1}{M_t} \sum_{k=1}^{M_t} \log f(\mathbf{y}, \mathbf{u}^{(t,k)}; \boldsymbol{\psi}). \tag{4.9}$$

The Monte Carlo EM algorithm uses $\tilde{Q}$ in place of Q; that is, the $(t+1)$st MCEM update, $\boldsymbol{\psi}^{(t+1)}$, satisfies

$$\tilde{Q}(\boldsymbol{\psi}^{(t+1)}|\boldsymbol{\psi}^{(t)}) \geq \tilde{Q}(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t)}), \ \forall \boldsymbol{\psi}.$$

Notice that in equation (4.9) we have explicitly allowed the Monte Carlo sample size, M_t, to depend on the iteration number t. The reason for this is that MCEM typically requires M_t to be increased in successive iterations in order for the sequence of estimates to converge. However, we will sometimes find it necessary

to consider MCEM with a constant Monte Carlo sample size per iteration. In this case, we will simply write M, omitting the subscript, with the understanding that the sample size remains fixed for all t.

We have pointed out that the EM update solves the estimating equations $\mathbf{0} = \mathbf{F}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)})$, with $\mathbf{F}$ defined in (3.15). A Monte Carlo approximation to $\mathbf{F}$ is straightforward. Let

$$\tilde{\mathbf{F}}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}) = \frac{1}{M_t} \sum_{k=1}^{M_t} \frac{\partial}{\partial \boldsymbol{\psi}} \log f(\mathbf{y}, \mathbf{u}^{(t,k)}; \boldsymbol{\psi}). \tag{4.10}$$

Then the $(t+1)$st MCEM update solves $\tilde{\mathbf{F}}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}) = \mathbf{0}$.

Notice that $\tilde{Q}$ in (4.9) and $\tilde{\mathbf{F}}$ in (4.10) are random variates which have expectation Q and $\mathbf{F}$, respectively. (The expectation taken with respect to the conditional density $g(\mathbf{u}|\mathbf{y}; \boldsymbol{\psi}^{(t)})$). Therefore one can typically decompose MCEM into two components: a *deterministic* component that follows the EM algorithm exactly, and a *stochastic* component due to Monte Carlo error. In other words, MCEM is a stochastic algorithm that follows, on average, the path of the underlying deterministic EM, but shows random variation about this path. Consider the following example for illustration.

Example: OWMM. Consider the OWMM from Section 2.1.3. We show in the Appendix (Section A.4, equation (A.22)), that the $(t+1)$st MCEM update is

$$\mu^{(t+1)} = \hat{\mu} + b(\mu^{(t)} - \hat{\mu}) + e_t, \tag{4.11}$$

where

$$e_t \sim \mathcal{N}(0, \sigma^2_{MC}/M) \text{ and } \sigma^2_{MC} = b\sigma_0^2/(mr) \tag{4.12}$$

and $b = \sigma_1^2/(\sigma_1^2 + \sigma_0^2/r)$. Notice that equation (4.11) equals the update of the deterministic EM algorithm in (3.19) plus Monte Carlo error, e_t. Moreover, since e_t has mean zero, MCEM is *on average* identical to EM. However, MCEM varies randomly about the EM path due to the error term e_t. Notice that the Monte

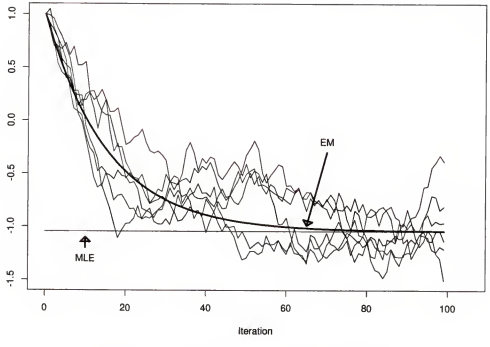

Figure 4.2: EM and MCEM in the OWMM.

Carlo error variance is proportional to $1/M$. Thus, for a *constant* Monte Carlo sample size M, the error variance of MCEM does not decrease and therefore the algorithm does *not* converge. This fact is illustrated graphically in Figure 4.2.

Figure 4.2 shows the first 100 iterations of the EM algorithm (thick line) for a simulated set of data from the OWMM. Notice that EM converges after about 80 iterations to the MLE. Figure 4.2 also shows 6 independent runs of MCEM (thin lines) with a constant Monte Carlo sample size per iteration. We observe that MCEM follows the path of EM on average, but individual runs vary randomly about this average path. In particular, notice that MCEM continues to show random variation, even after EM converges. Several authors (Wei and Tanner, 1990; McCulloch, 1997; Booth and Hobert, 1999) propose to increases M as the algorithm progresses. However, most of the suggestions of *how* to increase M are somewhat ad hoc and driven by the individual application of the algorithm. Booth

and Hobert (1999) are the first to develop an *automated* sample size updating scheme for the MCEM algorithm.

Automated MCEM. Booth and Hobert (1999) develop an MCEM algorithm that automatically increases M based on random samples from $g(\mathbf{u}|\mathbf{y};\boldsymbol{\psi}^{(t)})$. They show that using random sampling allows the Monte Carlo error to be assessed at each iteration using standard central limit theory combined with Taylor series methods. These can be used to construct a sandwich variance estimate for the maximizer at each approximate E-step. Specifically, let $\boldsymbol{\psi}^{(t+1)}$ denote the MCEM update which satisfies $\tilde{\mathbf{F}}(\boldsymbol{\psi},\boldsymbol{\psi}^{(t)}) = \mathbf{0}$ and define $\tilde{\mathbf{F}}^{(1)}(\mathbf{x},\mathbf{y}) \equiv \partial\tilde{\mathbf{F}}(\mathbf{x},\mathbf{y})/\partial\mathbf{x}$. If we let $\boldsymbol{\psi}^{*(t+1)}$ denote the value that satisfies $\mathbf{F}(\boldsymbol{\psi},\boldsymbol{\psi}^{(t)}) = \mathbf{0}$ (that is, $\boldsymbol{\psi}^{*(t+1)}$ is the true EM update from $\boldsymbol{\psi}^{(t)}$), then it follows that, conditional on the current iteration value $\boldsymbol{\psi}^{(t)}$, $\boldsymbol{\psi}^{(t+1)}$ is approximately normal distributed with mean $\boldsymbol{\psi}^{*(t+1)}$ and variance

$$\mathrm{Var}(\boldsymbol{\psi}^{(t+1)}|\boldsymbol{\psi}^{(t)}) \approx$$
$$\tilde{\mathbf{F}}^{(1)}(\boldsymbol{\psi}^{*(t+1)},\boldsymbol{\psi}^{(t)})^{-1}\mathrm{Var}[\tilde{\mathbf{F}}(\boldsymbol{\psi}^{*(t+1)},\boldsymbol{\psi}^{(t)})]\tilde{\mathbf{F}}^{(1)}(\boldsymbol{\psi}^{*(t+1)},\boldsymbol{\psi}^{(t)})^{-1} \tag{4.13}$$

A sandwich estimate of $\mathrm{Var}(\boldsymbol{\psi}^{(t+1)}|\boldsymbol{\psi}^{(t)})$ is obtained by substituting $\boldsymbol{\psi}^{(t+1)}$ in place of $\boldsymbol{\psi}^{*(t+1)}$ on the right-hand side of expression (4.13) and using the estimate

$$\widehat{\mathrm{Var}}\left(\tilde{\mathbf{F}}(\boldsymbol{\psi}^{*(t+1)},\boldsymbol{\psi}^{(t)})\right) = \frac{1}{M_t^2}\sum_{k=1}^{M_t}[\frac{\partial \log f(\mathbf{y},\mathbf{u}^{(t,k)};\boldsymbol{\psi}^{(t+1)})}{\partial\boldsymbol{\psi}}][\frac{\partial \log f(\mathbf{y},\mathbf{u}^{(t,k)};\boldsymbol{\psi}^{(t+1)})}{\partial\boldsymbol{\psi}}]'.$$

This suggests a rule for automatically increasing the Monte Carlo sample size after iterations in which the true EM step is swamped by Monte Carlo error. In particular, Booth and Hobert suggest that after the $(t+1)$st iteration, construct a $100(1-\alpha)\%$ confidence ellipsoid for the true (but unknown) EM step $\boldsymbol{\psi}^{*(t+1)}$ by using the approximate normality of its Monte Carlo estimate $\boldsymbol{\psi}^{(t+1)}$ together with the estimate of its approximate variance given in (4.13). If the previous Monte Carlo EM step, $\boldsymbol{\psi}^{(r)}$, lies in that region, then the EM step was swamped by Monte

Carlo error, and the Monte Carlo sample size M should be increased. Booth and Hobert suggest a rule of the form $M_{t+1} \leftarrow M_t(1+a)$, where $0 < a < 1$. In their examples they used $\alpha = 0.25$ and $a \in \{\frac{1}{3}, \frac{1}{4}, \frac{1}{5}\}$, but the optimal choice of α and a is a topic that needs further investigation.

Stopping Rules for Stochastic Algorithms. Booth and Hobert also address the issue of a stopping rule for stochastic algorithms. Deterministic algorithms, like NR or EM, typically stop when the relative change in the parameter values from successive iterations is small; that is, they use a stopping rule (or convergence criterion) of the form

$$\max_s \left(\frac{|\psi_s^{(t+1)} - \psi_s^{(t)}|}{|\psi_s^{(t)}| + \delta_1} \right) < \delta_2, \tag{4.14}$$

where δ_1 and δ_2 are predetermined constants. One problem with this stopping rule in the context of stochastic algorithms (like MCEM) is that $\boldsymbol{\psi}^{(t+1)}$ may be very close to $\boldsymbol{\psi}^{(t)}$ simply because of a large Monte Carlo error associated with $\boldsymbol{\psi}^{(t+1)}$ and *not* because the algorithm is close to convergence. To reduce the risk of stopping prematurely due to Monte Carlo error, Booth and Hobert suggest to stop the algorithm when (4.14) is satisfied, say, C consecutive times.

Example: OWMM. Figure 4.3 shows 6 runs of automated MCEM for a simulated set of data from the OWMM, each started from the same point, $\mu^{(0)} = 1$, using convergence parameters $\alpha = 0.25$, $a = 1/5$, $\delta_1 = 0.001$, $\delta_2 = 0.001$ and $C = 3$ and an initial Monte Carlo sample size of $M_0 = 10$. The first row in Figure 4.3 (plots 1 through 3) shows the MCEM histories for each of the 6 runs. Plot 1 shows the *entire* history, whereas plots 2 and 3 only show *windows* of the first and final 50 iterations, respectively. The second row in Figure 4.3 (plots 4 through 6) shows the corresponding sample size.

Observe that the variability of the MCEM estimates decreases as the algorithm progresses and each of the 6 runs converges between iteration 80 and 100. However, M_t increases very rapidly at the later stages of the algorithm, reaching values

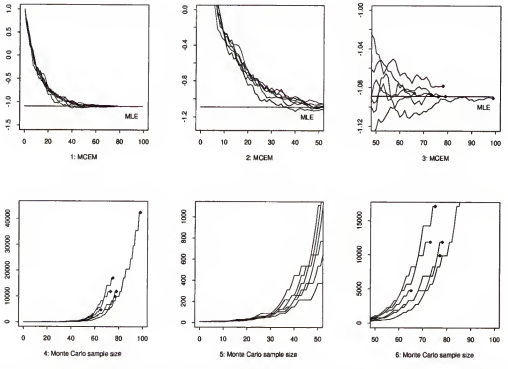

Figure 4.3: Automated MCEM and Monte Carlo sample sizes in the OWMM.

between 10,000 and 40,000 at the final iterations. This indicates that extremely large Monte Carlo sample sizes are needed for convergence of MCEM. On the other hand, hardly any of the simulation effort is spent at the early iterations. Indeed, plots 5 and 6 show that the Monte Carlo sample sizes barely exceed 1,000 in the first 50 iterations.

4.3.2 Monte Carlo Newton-Raphson (MCNR)

Recall that Newton-Raphson requires evaluating the score function $\mathbf{S}$ and the Hessian $\mathbf{H}$. However, we have argued at the beginning of this chapter that $\mathbf{S}$ (and therefore also $\mathbf{H}$) are typically intractable in GHMs. One solution is to approximate $\mathbf{S}$ and $\mathbf{H}$ using Monte Carlo integration. This leads to the Monte Carlo Newton-Raphson algorithm.

A suitable Monte Carlo approximation to $\mathbf{S}$ is suggested by equation (3.26) which provides a representation for the score function of the form $\mathbf{S}(\boldsymbol{\psi}) = E[\partial \log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi}) / \partial \boldsymbol{\psi} | \mathbf{y}; \boldsymbol{\psi}]$. Therefore, if $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(M_t)}$ denotes a random sample from $g(\mathbf{u}|\mathbf{y}; \boldsymbol{\psi})$, then a Monte Carlo approximation to $\mathbf{S}$ is given by

$$\tilde{\mathbf{S}}(\boldsymbol{\psi}) = \frac{1}{M_t} \sum_{k=1}^{M_t} \frac{\partial}{\partial \boldsymbol{\psi}} \log f(\mathbf{y}, \mathbf{u}^{(k)}; \boldsymbol{\psi}). \tag{4.15}$$

A Monte Carlo approximation to the Hessian matrix $\mathbf{H}$ can be found in similar manner. We have already shown (see equation (3.27)) that

$$\mathbf{H}(\boldsymbol{\psi}) = \mathbf{H}_Q(\boldsymbol{\psi}) + \mathrm{Var}\left(\frac{\partial}{\partial \boldsymbol{\psi}} \log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\psi}) \middle| \mathbf{y}; \boldsymbol{\psi} \right), \tag{4.16}$$

where $\mathbf{H}_Q(\boldsymbol{\psi})$ is the Hessian of the Q-function in equation (3.23). Thus, a Monte Carlo approximation to the first term on the right hand side of (4.16) is

$$\tilde{\mathbf{H}}_Q(\boldsymbol{\psi}) = \frac{1}{M_t} \sum_{k=1}^{M_t} \frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}'} \log f(\mathbf{y}, \mathbf{u}^{(k)}; \boldsymbol{\psi}). \tag{4.17}$$

Similarly, an approximation to the second term on the right hand side of (4.16) is

$$\tilde{\mathbf{V}}(\boldsymbol{\psi}) = \frac{1}{M_t} \sum_{k=1}^{M_t} [\frac{\partial \log f(\mathbf{y}, \mathbf{u}^{(k)}; \boldsymbol{\psi})}{\partial \boldsymbol{\psi}} - \tilde{\mathbf{S}}(\boldsymbol{\psi})][\frac{\partial \log f(\mathbf{y}, \mathbf{u}^{(k)}; \boldsymbol{\psi})}{\partial \boldsymbol{\psi}} - \tilde{\mathbf{S}}(\boldsymbol{\psi})]'. \tag{4.18}$$

Therefore, we can approximate $\mathbf{H}$ by

$$\tilde{\mathbf{H}}(\boldsymbol{\psi}) = \tilde{\mathbf{H}}_Q(\boldsymbol{\psi}) + \tilde{\mathbf{V}}(\boldsymbol{\psi}). \tag{4.19}$$

Equation (4.19) offers a very appealing approximation to $\mathbf{H}$ based on the representation in (4.16); however, other approximations are possible. Some of these are discussed later in this section.

The Monte Carlo Newton-Raphson algorithm uses $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{H}}$ in place of $\mathbf{S}$ and $\mathbf{H}$. Thus, MCNR updates the parameter estimate according to

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} - \tilde{\mathbf{H}}(\boldsymbol{\psi}^{(t)})^{-1} \tilde{\mathbf{S}}(\boldsymbol{\psi}^{(t)}). \tag{4.20}$$

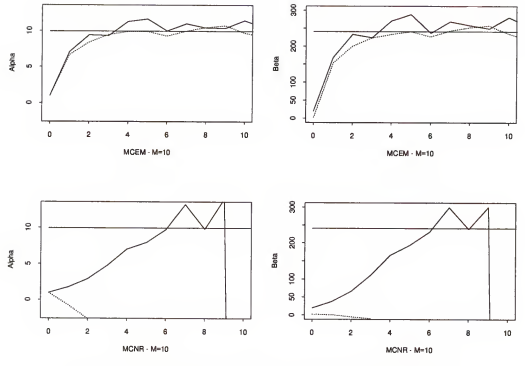

Figure 4.4: MCEM and MCNR in beta-binomial model.

Like MCEM, this algorithm also does not converge unless the Monte Carlo sample size is increased as the algorithm progresses.

Example: Beta-Binomial Model. Recall the performance of the deterministic algorithms, EM and NR, for the beta-binomial model: EM converged for both of the starting values, $(1;2)$ and $(1;20)$, whereas NR only converged for the latter value (see Figures 3.1 and 3.3). Figure 4.4 shows the behavior of the corresponding Monte Carlo algorithms with constant Monte Carlo sample sizes, $M = 10$. The dashed and solid lines correspond to the starting values $(1;2)$ and $(1;20)$, respectively. MCEM performs as expected: it reaches the neighborhood of the MLE for both of the starting values and oscillates randomly about the MLE from there on. MCNR diverges for the value $(1;2)$, which is not surprising as this value also caused convergence problems for the deterministic NR. However, the starting value $(1;20)$ creates unforeseen problems for MCNR. For this value, MCNR

performs as expected initially. It reaches the neighborhood of the MLE quickly and then fluctuates randomly about this point for the next few iterations. However, the algorithm then breaks down and diverges to the boundary of the parameter space.

Modifications of MCNR. The instability of the Monte Carlo Hessian, $\tilde{\mathbf{H}}$, in (4.19) is the main reason for the erratic behavior of MCNR. If M is not very large, $\tilde{\mathbf{H}}$ can vary dramatically from one iteration to the next. Since $\tilde{\mathbf{H}}$ governs the step size, such erratic behavior can result in huge jumps in the parameter estimates. In some cases the estimates may jump outside the domain of attraction, causing MCNR to diverge. We have found that the following modifications of MCNR often work well in practice.

1. Use $\tilde{\mathbf{H}}_Q$

   We have argued in Section 3.5 that $\mathbf{H}(\boldsymbol{\psi}) \leq \mathbf{H}_Q(\boldsymbol{\psi})$ (in the semi-positive definite order), causing Lange's modified EM algorithm to take smaller steps than NR. This implies that replacing $\tilde{\mathbf{H}}$ by $\tilde{\mathbf{H}}_Q$ in the corresponding Monte Carlo version of the algorithm will lead to smaller steps sizes for MCNR and, thus, to a more stable algorithm. Notice, however, that this approach results in a modified MCEM algorithm, rather than a NR-type method.

2. Use $\bar{\mathbf{H}}_t$

   A second modification is to use an *average* of Monte Carlo Hessian matrices; in particular, if we let $\tilde{\mathbf{H}}_i = \tilde{\mathbf{H}}(\boldsymbol{\psi}^{(i)})$ be the value of $\tilde{\mathbf{H}}$ evaluated at the ith parameter update, then in the $(t+1)$st iteration of MCNR we replace $\tilde{\mathbf{H}}(\boldsymbol{\psi}^{(t)})$ by the average

   $$\bar{\mathbf{H}}_t = \frac{1}{t+1} \sum_{i=0}^{t} \tilde{\mathbf{H}}_i. \tag{4.21}$$

   This approach smoothes the changes in the Hessian matrix between successive iterations. However, recall that, compared to EM, NR takes rather small steps at the early iterations (see Figure 3.3, for example). Thus, using an

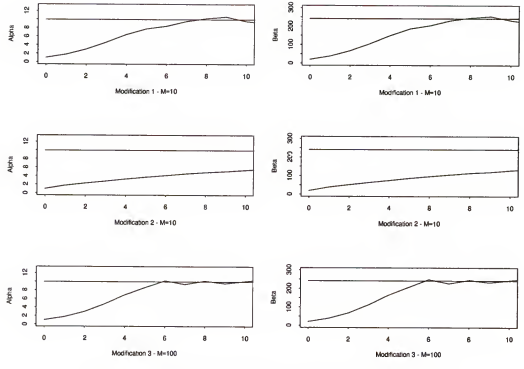

Figure 4.5: Modified MCNR in beta-binomial model.

average of Hessians may lead to smaller step sizes at later iterations and thus to a slower convergence rate for MCNR.

3. Increase M

   The variability of $\tilde{\mathbf{H}}$ can be decreased simply by using a larger Monte Carlo sample size. However, using a larger M may decrease the overall efficiency of MCNR measured by total amount of simulation or by total computing time needed for the algorithm to converge.

Example: Beta-Binomial Model. Figure 4.5 illustrates the performance of these three modifications for the beta-binomial model with the starting value $(1; 20)$. Notice that each of the three modifications succeeds in stabilizing MCNR. However, using an average of Hessians, $\bar{\mathbf{H}}_t$, slows down the algorithm significantly. With modifications 1 and 3, MCNR reaches the neighborhood of the MLE after about six to eight iterations, whereas using $\bar{\mathbf{H}}_t$, it takes significantly longer.

4.4 Stochastic Approximation (SA)

In Section 4.3 we introduced MCEM and MCNR, two iterative methods useful for approximating the MLE. Both of these methods require that the Monte Carlo sample size M be increased successively for convergence. In this section we consider methods that converge for constant M. These methods combine the concepts of EM and NR together with those of stochastic approximation.

The origin of stochastic approximation (SA) dates back to the work of Robbins and Monro (1951). It is for this reason that SA (and its variants) can often be found in the literature under the name "Robbins-Monro" procedure.

Suppose $S(\psi)$ is a function with scalar argument ψ. Suppose further that we want to find the root, $\hat{\psi}$, of S satisfying $S(\hat{\psi}) = 0$. If S is known, deterministic methods like Newton-Raphson can be used to approximate $\hat{\psi}$. Robbins and Monro consider a stochastic generalization of the above problem in which the precise form of the function S is unknown. Instead of observing $S(\psi)$ exactly, suppose we only observe a random variable $V(\psi)$ with distribution function $Pr(V(\psi) \leq v) = G(v|\psi)$, such that

$$S(\psi) = \int v \; dG(v|\psi). \tag{4.22}$$

Notice that in GHMs, $S(\psi)$ can be considered the score function and equation (3.26) reveals that (4.22) holds with $V(\psi) \equiv \partial \log f(\mathbf{y}, \mathbf{u}; \psi)/\partial\psi$ and $dG(v|\psi) \equiv g(\mathbf{u}|\mathbf{y}; \psi)d\mathbf{u}$. Robbins and Monro (1951) show that for a sequence of positive weights, $\{\gamma_t\}_{t\geq 0}$, such that

$$\sum_t \gamma_t = \infty \quad \text{and} \quad \sum_t \gamma_t^2 < \infty, \tag{4.23}$$

the Markov chain, $\{\psi^{(t)}\}_{t\geq 0}$, defined by

$$\psi^{(t+1)} = \psi^{(t)} - \gamma_t v_t \tag{4.24}$$

converges in probability to the solution $\hat{\psi}$, where v_t satisfies $Pr(v_t \leq v|\psi^{(t)}) = G(v|\psi^{(t)})$.

Many early refinements of the work of Robbins and Monro exist, most of which modify the convergence details of $\psi^{(t)} \to \hat{\psi}$ (see, e.g., Blum, 1954; Kallianpur, 1954; Wolfowitz, 1956). Kiefer and Wolfowitz (1952) develop a modification of SA, suitable to find the *maximum* of the function $S(\psi)$. Kesten (1958) derives an *accelerated* version of SA, which takes into account the frequency of fluctuations in the sign of two successive iterations, $\text{sign}(\psi^{(t)} - \psi^{(t-1)})$. More recently, SA with averaging of the estimates, $\bar{\psi}_t = \sum_{i=0}^{t} \psi^{(i)}/(t+1)$, has been proposed as a further way of accelerating the algorithm (see, e.g., Györfi and Walk, 1996; Wang et al., 1997; Tang et al., 1999). Kushner (1987) studies a combination of SA and simulated annealing in order to overcome local solutions. Many more extensions and modifications of SA exist.

Ruppert (1985) establishes the following similarity between SA and Newton-Raphson. Let $\mathbf{S}(\boldsymbol{\psi})$ be a function from $\mathcal{R}^S$ to $\mathcal{R}^S$ and $\boldsymbol{\psi} \in \mathcal{R}^S$. Ruppert derives a multivariate version of SA [1],

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} - \frac{A}{t+1} \tilde{\mathbf{H}}_t^{-1} \tilde{\mathbf{S}}_t, \tag{4.25}$$

where $A > 0$ and $\tilde{\mathbf{S}}_t$ and $\tilde{\mathbf{H}}_t$ are (stochastic) estimates for $\mathbf{S}(\boldsymbol{\psi}_t)$ and its gradient, respectively. Notice that omitting the down-weighting factor $A/(t+1)$ and replacing $\tilde{\mathbf{S}}_t$ and $\tilde{\mathbf{H}}_t$ in (4.25) by the score function $\mathbf{S}$ and the Hessian $\mathbf{H}$ results in the Newton-Raphson algorithm. Equation (4.25) is the starting point for our discussion in the next section.

[1] Ruppert actually considers a version of SA suitable to find the *maximum* of the function $\mathbf{S}(\boldsymbol{\psi})$; however, it is straightforward to apply his ideas to the case $\mathbf{S}(\boldsymbol{\psi}) = \mathbf{0}$.

4.4.1 Stochastic Approximation Newton-Raphson (SANR)

Let $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(M)}$ be a random sample from $g(\mathbf{u}|\mathbf{y}; \boldsymbol{\psi}^{(t)})$. The SANR algorithm updates the parameter estimate according to

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} - \gamma_t \, \tilde{\mathbf{H}}(\boldsymbol{\psi}^{(t)})^{-1} \tilde{\mathbf{S}}(\boldsymbol{\psi}^{(t)}), \tag{4.26}$$

where $\{\gamma_t\}_{t \geq 0}$ is a sequence of decreasing weights satisfying (4.23) and $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{S}}$ are the Monte Carlo Hessian and score function in (4.19) and (4.15), respectively. Notice that the Monte Carlo sample size M is typically constant over all iterations and has to be chosen before starting the algorithm.

Gu and Li (1998) use a special case of the algorithms in (4.25) and (4.26) with the specific weight, $\gamma_t = 1/(1+t)$. Gu and Li (1998) also propose different choices for the Monte Carlo Hessian $\tilde{\mathbf{H}}$. They suggest using $\tilde{\mathbf{H}}_Q$ in (4.17) at the early iterations of SANR and switching to $\tilde{\mathbf{H}}_t$ in (4.21) when "t is large". This recommendation is in line with the comments made in Section 4.3.2 concerning the stability of MCNR.

Notice that the difference between SANR and MCNR is only due to the down-weighting factor γ_t. However, this down-weighting factor changes the properties of the algorithm significantly. Consider the following example for illustration.

Example: OWMM. Consider the OWMM and assume that the Monte Carlo sample size M is constant for all iterations. Let us consider MCNR first. We show in the Appendix (Section A.4, equation (A.23)) that the Monte Carlo score function is

$$\tilde{S}(\mu) = \frac{mr}{\sigma_0^2}[(1-b)(\hat{\mu} - \mu) + e_t], \tag{4.27}$$

with e_t defined in equation (4.12). Recall that the (exact) Hessian for this model is $H = -(1-b)mr/\sigma_0^2$ (see equation (3.10)). Using H (instead of an approximation

$\tilde{H}$), the difference between the $(t+1)$st MCNR update and the MLE is

$$\mu^{(t+1)} - \hat{\mu} = e^*, \tag{4.28}$$

where $e^* \sim N(0, (1-b)^{-2}\sigma^2_{MC}/M)$ and σ^2_{MC} defined in (4.12). Notice that the $(t+1)$st iteration does not improve over the tth iteration in the sense of a reduced variance; in fact, for constant M, the variance of the MCNR estimate does not decrease for successive iterations and therefore MCNR does not converge.

Now consider SANR in (4.26) with the weight $\gamma_t = 1/(1+t)$. We, again, use $\tilde{S}$ in (4.27) and the exact Hessian H instead of an approximation $\tilde{H}$. The $(t+1)$st SANR update is

$$\mu^{(t+1)} = \mu^{(t)} + \frac{1}{t+1}(\hat{\mu} - \mu^{(t)} + e^*). \tag{4.29}$$

It follows from equation (4.29) that the difference between the $(t+1)$st SANR update and the MLE is

$$\mu^{(t+1)} - \hat{\mu} = \frac{1}{t+1}\sum_{i=1}^{t+1} e^* \equiv e^{**}, \tag{4.30}$$

say, where $e^{**} \sim N(0, (1-b)^{-2}\sigma^2_{MC}/[M(t+1)])$. Thus, the variance of e^{**} decreases as the number of iterations, t, increases. This implies that SANR converges for a constant Monte Carlo sample size M. Figure 4.6 illustrates this point.

Figure 4.6 shows the first 400 iterations of MCNR and SANR in the OWMM, generated according to equations (4.28) and (4.29), respectively. The MLE is indicated by the horizontal line in each plot. The first plot in Figure 4.6 shows the iteration history for MCNR. Clearly, MCNR does not converge and continues to fluctuate randomly around the MLE even after 400 iterations. The second plot shows SANR. Observe, that SANR shows some variability in the initial iterations. However, the variability decreases rapidly and SANR converges to the MLE after about 350 iterations.

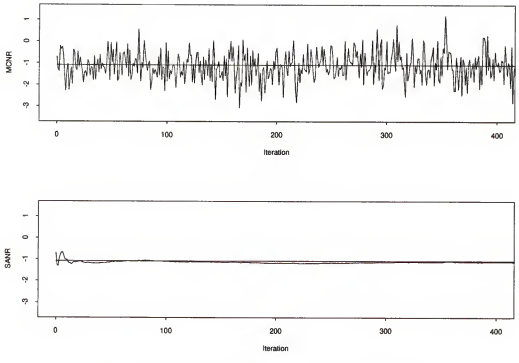

Figure 4.6: MCNR and SANR in the OWMM.

Convergence Rate of Stochastic Approximation. A word of caution is necessary here. In more general models, one typically has to pay a price for convergence with constant M. And this price is often an extremely slow rate of convergence. Consider the next example for illustration.

Example: Beta-Binomial Model. Figure 4.7 illustrates the performance of SANR in the beta-binomial model for different weights of the form

$$\gamma_t = \frac{A}{A+t}, \; t = 0, 1, 2, \ldots. \tag{4.31}$$

For four different values of A we ran SANR for 200 iterations, starting each run from $(\alpha^{(0)}; \beta^{(0)}) = (1; 20)$, using the Monte Carlo Hessian $\tilde{\mathbf{H}}$ in (4.19) and a constant Monte Carlo sample size, $M = 100$. The plots show only the iteration histories for the α-component (thick lines). The behavior of the β-component, however, is very similar. For comparison, we also included the iteration history for

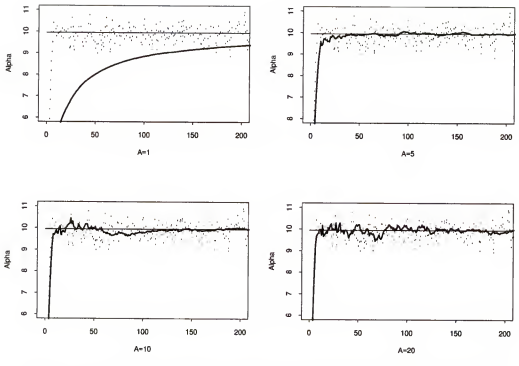

Figure 4.7: Rate of convergence of SANR in the beta-binomial model.

one run of MCNR (thin, dotted lines) from the same starting value with constant $M = 100$.

Notice that for the value $A = 1$ the convergence rate of SANR is extremely slow; even after 200 iterations the algorithm is still far from the MLE. However, there is barely any random noise in the parameter updates in this case. Conversely, as A increases, the Monte Carlo error grows but, at the same time, the rate of convergence of the algorithm improves. In particular, as A gets larger, SANR resembles more and more MCNR. This fact reveals the conflict that the user of SANR (and also other stochastic approximation methods) often faces. If he chooses weights, γ_t, that are very small at the beginning of the algorithm, then he sacrifices a fast convergence rate for almost no Monte Carlo error. Conversely, if he chooses weights that are large at the initial stages, then the algorithm reaches a neighborhood of the MLE quickly at the price of a possibly huge Monte Carlo

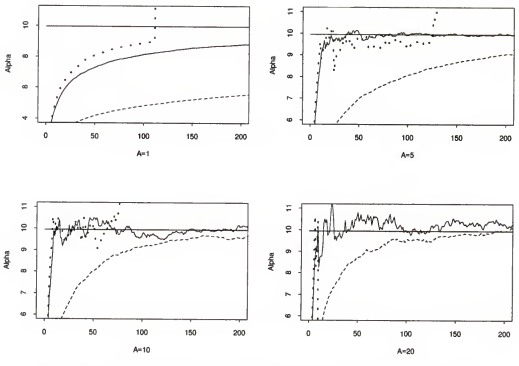

Figure 4.8: Modifications of SANR in the beta-binomial model.

error. Clearly, the best choice of γ_t finds a good compromise between a fast convergence rate and small Monte Carlo error variance. However, no general guidelines for the choice of γ_t exist. In fact, most recommendations are somewhat ad hoc and tailored towards a specific model and set of data.

Modifications of SANR. We pointed out in Section 4.3.2 that MCNR can have stability problems when using $\tilde{\mathbf{H}}$ in (4.19) together with a, typically, small Monte Carlo sample size. These problems carry over to SANR. Observe that in Figure 4.7 we used SANR with $M = 100$. Often, however, it is desirable to use SANR with a *smaller* M in order to increase the efficiency of the algorithm. Figure 4.8 shows 200 iterations of SANR with $M = 10$ using (1) $\tilde{\mathbf{H}}$ (thick dotted lines); (2) $\tilde{\mathbf{H}}_Q$ in (4.17) (thin solid lines); and (3) $\tilde{\mathbf{H}}_t$ in (4.21) (thin dashed lines).

We observe that SANR breaks down when using $\tilde{\mathbf{H}}$. This implies that an alternative to $\tilde{\mathbf{H}}$ should be used when the objective is to run SANR with a small

M. Figure 4.8 shows that both of the alternatives, $\tilde{\mathbf{H}}_Q$ as well as $\tilde{\mathbf{H}}_t$, stabilize the algorithm. Notice, however, that using $\tilde{\mathbf{H}}_t$ in place of $\tilde{\mathbf{H}}$ slows down the algorithm significantly, especially when $A = 1$, where an already very slow rate of convergence for SANR is drastically reduced by using $\tilde{\mathbf{H}}_t$. On the other hand, using $\tilde{\mathbf{H}}_Q$ in place of $\tilde{\mathbf{H}}$ works quite well and the results are very similar to those in Figure 4.7 (with a larger M). However, as with MCNR, replacing $\tilde{\mathbf{H}}$ by $\tilde{\mathbf{H}}_Q$ results in an EM-type algorithm which is considered in detail in the next section.

4.4.2 Stochastic Approximation EM (SAEM)

Delyon et al. (1999) propose a stochastic approximation to the EM algorithm. They suggest replacing the $\tilde{Q}$-function in the MCEM algorithm with a weighted average of $\tilde{Q}$-like functions. Specifically, the $(t+1)$st update is obtained by maximizing

$$\bar{Q}_{t+1}(\boldsymbol{\psi}) = (1-\gamma_t)\bar{Q}_t(\boldsymbol{\psi}) + \gamma_t \tilde{Q}(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t)}), \tag{4.32}$$

with $\tilde{Q}$ defined in equation (4.9) and $\{\gamma_t\}_{t\geq 0}$, a sequence of decreasing weights that satisfy (4.23). In general, $\gamma_0 = 1$ and $\bar{Q}_0 = 0$, which implies that the first SAEM update is identical to the first MCEM update. In fact, if $\gamma_t \equiv 1, \forall t$, SAEM reduces exactly to MCEM. The $(t+1)$st SAEM update is typically obtained by solving the estimating equation $\bar{\mathbf{F}}_{t+1}(\boldsymbol{\psi}) = \mathbf{0}$, where

$$\bar{\mathbf{F}}_{t+1}(\boldsymbol{\psi}) = (1-\gamma_t)\bar{\mathbf{F}}_t(\boldsymbol{\psi}) + \gamma_t \tilde{\mathbf{F}}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}), \tag{4.33}$$

and $\tilde{\mathbf{F}}$ is defined in (4.10). Delyon et al. establish that SAEM converges to a local maximum of the likelihood function for a fixed Monte Carlo sample size, M.

Example: OWMM. Consider the OWMM and assume that the Monte Carlo sample size M is fixed for all iterations. It follows from equation (4.11) that the difference between the $(t+1)$st MCEM update and the MLE is given by

$$\mu^{(t+1)} - \hat{\mu} = b^{t+1}(\mu^{(0)} - \hat{\mu}) + e^{\dagger}, \tag{4.34}$$

where

$$e^{\dagger} = \sum_{i=0}^{t} b^{t-i} e_i \sim N(0, \frac{\sigma_{MC}^2}{M} \sum_{i=0}^{t} b^{2(t-i)}) \tag{4.35}$$

with e_i and σ_{MC}^2 defined in (4.12). Notice that $b^{t+1}(\mu^{(0)} - \hat{\mu}) \to 0$ (as $t \to \infty$) since $0 \leq b < 1$. However, the variance of $e^{\dagger}$ in (4.35) does not decrease for constant M and hence MCEM does not converge for fixed Monte Carlo sample size.

Now consider SAEM. One can show (see Section 7.1 for more details) that for weights of the form $\gamma_t = 1/(1+t)$, the difference between the $(t+1)$st SAEM update and the MLE is

$$\mu^{(t+1)} - \hat{\mu} = \bar{b}_{t+1}(\mu^{(0)} - \hat{\mu}) + e^{\ddagger} \tag{4.36}$$

where

$$\bar{b}_{t+1} = \prod_{i=0}^{t} \frac{b+i}{1+i} = \frac{\Gamma(t+1+b)}{\Gamma(t+2)\Gamma(b)} \tag{4.37}$$

$$e^{\ddagger} = \frac{1}{t+1} \sum_{i=0}^{t} e_i \sim N(0, \frac{\sigma_{MC}^2}{M(t+1)}). \tag{4.38}$$

Notice that for large t, $\Gamma(t+1+b)/\Gamma(t+2) \sim t^{b-1}$ (see Abramowitz and Stegun, 1992, p.257). Thus, since $0 \leq b < 1$, $\bar{b}_{t+1} \to 0$ (as $t \to \infty$). Moreover, the variance of $e^{\ddagger}$ vanishes as t increases. Thus, in contrast to MCEM, SAEM converges for a constant Monte Carlo sample size.

Example: Beta-Binomial Model. Figure 4.9 illustrates SAEM's convergence rate for the beta-binomial model with weights, γ_t, of the form (4.31). The plots show the first 200 iterations of SAEM (thick lines) run with a constant Monte Carlo sample size, $M = 10$, per iteration, for $A \in \{1; 5; 10; 20\}$. For comparison, we included MCEM (thin, dotted lines) run with the same constant M. Clearly, the smaller the weights are at the early stages of the algorithm, the faster SAEM reduces Monte Carlo error. However, at the same time, the algorithm's convergence rate deteriorates significantly with smaller values of A. As with SANR, there is

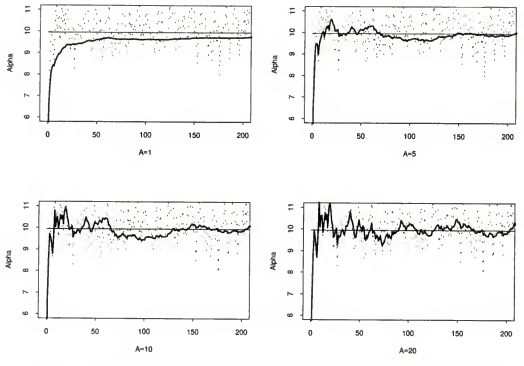

Figure 4.9: Rate of convergence of SAEM in the beta-binomial Model

a trade-off between a fast rate of convergence and an efficient Monte Carlo error reduction.

Stopping Rules for Stochastic Approximation Algorithms. The appeal of stochastic approximation methods (like SANR or SAEM) is that they reduce Monte Carlo error and thus converge with constant Monte Carlo sample size. We have seen that a fast error reduction typically results in a very slow rate of convergence. Thus, differences between successive parameter updates are very small even when the algorithm is far from convergence. Since most common stopping rules are based on the detection of small differences between successive iterations, the application of these rules to stochastic approximation algorithms will often lead to a premature decision to stop.

Figure 4.10 shows what happens when applying the stopping rule in (4.14) (using the parameters $\delta_1 = 0.001$, $\delta_2 = 0.001$ and $C = 3$) to SAEM. The end

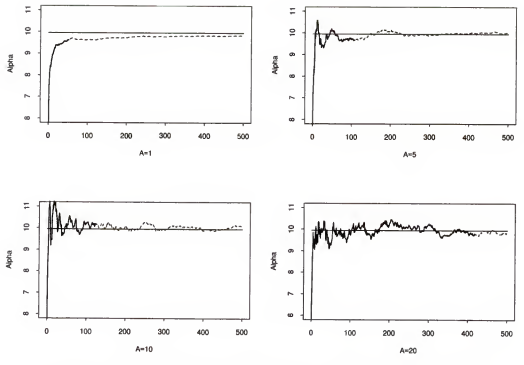

Figure 4.10: Convergence of SAEM in the beta-binomial Model

of the solid lines indicate the points at which SAEM stops due to (4.14). The dashed lines indicate the continuation of SAEM, had the stopping rule not been used. We observe that for each of the four runs, SAEM stops prematurely; that is, SAEM clearly stops far from the MLE and before eliminating the Monte Carlo error entirely.

Figure 4.10 clearly shows that stopping rules of the form (4.14) are inappropriate for stochastic approximation procedures. However, it is not clear at all whether there exist alternative rules, suitable for slow-converging algorithms like SAEM. In fact, it is very common to base the decision to stop on an inspection of a plot of the parameter updates versus the iteration number. However, this approach certainly does not automate the implementation of these methods and, furthermore, since stochastic approximation procedures tend to have very flat iteration histories, it will also often lead to wrong conclusions.

CHAPTER 5
EFFICIENCY OF MONTE CARLO EM AND SIMULATED MAXIMUM LIKELIHOOD

Unlike analytical or numerical approximation, the error of Monte Carlo based approximation methods can typically be made arbitrarily small by simply increasing the Monte Carlo sample size. This implies that the researcher can, at least in principle, control the accuracy of the estimates by choosing the total simulation amount appropriately. However, in practice this approach is often not feasible. Time limitations and computing power often restrict the total number of simulations, making it desirable to use the available simulations most efficiently. In particular, having several competing estimation methods to choose from, it is desirable to select that method that estimates the parameter most efficiently.

In this chapter we investigate the efficiency of MCEM and SML. The efficiency of a Monte Carlo method can be measured by the magnitude of its Monte Carlo error variance. A method with a large variance will need a larger total simulation amount to reach the same accuracy as a method with a smaller variance. McCulloch (1997, p.165) conducted an empirical study to investigate the variance of MCEM and SML in the case of the logistic-normal model. He found that SML "performs poorly [...], showing a very large variance" compared to MCEM. In this chapter, we investigate the Monte Carlo error of MCEM and SML analytically. In particular, using first and second order approximations we derive the Monte Carlo errors of MCEM and SML for the class of GLMMs. We show that the variance of SML is unbounded relative to that of MCEM and use this result to conclude that MCEM is a more efficient method in many applications of GLMMs.

This chapter is organized as follows. In Section 5.1 we derive the asymptotic Monte Carlo standard errors of the MCEM and SML estimates and discuss the implications of the results. In Section 5.2 we illustrate our results with two examples. We conclude this chapter by pointing out practical limitations to the use of SML.

5.1 Efficiency of MCEM and SML in GLMMs

5.1.1 Asymptotic Monte Carlo Error

Consider the GLMM defined in Section 2.1.1. We will assume that the variance components σ_0^2 and $\boldsymbol{\sigma}_1^2$ are known and that we are only interested in estimating $\boldsymbol{\beta}$ (i.e. $\boldsymbol{\psi} \equiv \boldsymbol{\beta}$). This assumption favors SML since the importance sampling distribution is exact and approximation (4.8) can be used. Suppressing the dependence on σ_0^2 and $\boldsymbol{\sigma}_1^2$, we will write $f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta}) \equiv f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta},\sigma_0^2)$ and $g(\mathbf{u}) \equiv g(\mathbf{u};\boldsymbol{\sigma}_1^2)$ for the conditional density of the data and the marginal density of the random effects, respectively.

In the following we derive the asymptotic distribution of a one-step MCEM iteration at the MLE, that is, given that the current parameter estimate equals the MLE, we derive the asymptotic distribution of the MCEM update. Let $\boldsymbol{\beta}^{(t)}$ be the tth MCEM update and $\boldsymbol{\beta}^{*(t)}$ the corresponding deterministic EM update. Notice that EM does not move from the MLE, $\hat{\boldsymbol{\beta}}$. Therefore, if $\boldsymbol{\beta}^{(t-1)} = \hat{\boldsymbol{\beta}}$, then $\boldsymbol{\beta}^{*(t)} = \hat{\boldsymbol{\beta}}$, and the difference, $\boldsymbol{\beta}^{(t)} - \hat{\boldsymbol{\beta}}$, is pure Monte Carlo error. Without loss of generality we may take $t = 1$. Notice that $\bar{\mathbf{F}}(\boldsymbol{\beta}, \hat{\boldsymbol{\beta}})$ in (4.10) is an average of the i.i.d. variates $X_k(\boldsymbol{\beta}) = \partial \log f(\mathbf{y}, \mathbf{u}^{(1,k)};\boldsymbol{\beta})/\partial\boldsymbol{\beta}$ with mean $E[X_1(\hat{\boldsymbol{\beta}})|\mathbf{y};\hat{\boldsymbol{\beta}}] = \mathbf{S}(\hat{\boldsymbol{\beta}}) = \mathbf{0}$ and variance $\mathrm{Var}[X_1(\hat{\boldsymbol{\beta}})|\mathbf{y};\hat{\boldsymbol{\beta}}] = \hat{\mathbf{I}}$, where

$$\hat{\mathbf{I}} = E\left[\left(\frac{\partial}{\partial\boldsymbol{\beta}}\log f(\mathbf{y},\mathbf{u};\boldsymbol{\beta})\right)\left(\frac{\partial}{\partial\boldsymbol{\beta}}\log f(\mathbf{y},\mathbf{u};\boldsymbol{\beta})\right)'\middle|\mathbf{y};\boldsymbol{\beta}\right]\Bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}. \tag{5.1}$$

Thus by the Central Limit Theorem, $\sqrt{M}\ \tilde{\mathbf{F}}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\beta}}) \rightarrow \mathcal{N}(\mathbf{0}, \tilde{\mathbf{I}})$. Notice furthermore that $\tilde{\mathbf{F}}(\boldsymbol{\beta}^{(1)}, \hat{\boldsymbol{\beta}}) = 0$. A first order Taylor expansion of $\tilde{\mathbf{F}}(\boldsymbol{\beta}^{(1)}, \hat{\boldsymbol{\beta}})$ about $\hat{\boldsymbol{\beta}}$ together with the Law of Large Numbers and Slutzky's Theorem reveals that

$$\Delta_{\text{MCEM}} \equiv \sqrt{M}(\boldsymbol{\beta}^{(1)} - \hat{\boldsymbol{\beta}}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\tau}^2_{\text{MCEM}}), \text{ as } M \rightarrow \infty, \tag{5.2}$$

where $\boldsymbol{\tau}^2_{\text{MCEM}} = \hat{\mathbf{J}}^{-1}\tilde{\mathbf{I}}\hat{\mathbf{J}}^{-1}$ and

$$\hat{\mathbf{J}} = E\left[\frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}\log f(\mathbf{y}, \mathbf{u}; \boldsymbol{\beta})\middle| \mathbf{y}; \boldsymbol{\beta}\right]\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}. \tag{5.3}$$

Notice that $\boldsymbol{\tau}^2_{\text{MCEM}}$ quantifies the Monte Carlo error of MCEM near the MLE.

In similar fashion, a first order Taylor expansion of $\partial\tilde{L}(\boldsymbol{\beta}|\mathbf{y})/\partial\boldsymbol{\beta}$ in (4.8) about $\hat{\boldsymbol{\beta}}$ shows that

$$\Delta_{\text{SML}} \equiv \sqrt{M}(\tilde{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\tau}^2_{\text{SML}}), \text{ as } M \rightarrow \infty, \tag{5.4}$$

with $\boldsymbol{\tau}^2_{\text{SML}} = \tilde{\mathbf{J}}^{-1}\tilde{\mathbf{I}}\tilde{\mathbf{J}}^{-1}$ and

$$\tilde{\mathbf{I}} = E\left[\left(\frac{\partial}{\partial\boldsymbol{\beta}}f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta})\right)\left(\frac{\partial}{\partial\boldsymbol{\beta}}f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta})\right)'\right]\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} \tag{5.5}$$

$$\tilde{\mathbf{J}} = E\left[\frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta})\right]\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}, \tag{5.6}$$

where the expectations in (5.5) and (5.6) are with respect to the marginal random effects density, $g(\mathbf{u})$.

Let $\mathbf{W}$ be the diagonal matrix of iterative GLM weights, $w_{ii} = 1/\{a_i V(\mu_i)\ g'(\mu_i)^2\}$, where $V(\mu_i) = b''(\theta_i)$ is the variance function for (2.1), and let $\hat{\mathbf{W}}$ be the value of $\mathbf{W}$ evaluated at the mode, $\hat{\mathbf{u}}$, of $f(\mathbf{y}, \mathbf{u}; \hat{\boldsymbol{\beta}})$. We show in the Appendix (Section A.5) that $|\boldsymbol{\tau}^2_{\text{MCEM}}|$, the *generalized variance* of Δ_{MCEM}, is proportional to $|\boldsymbol{\Sigma}|$, where

$$\boldsymbol{\Sigma} = [\mathbf{Z}'\hat{\mathbf{W}}\mathbf{Z} + \mathbf{G}^{-1}]^{-1}. \tag{5.7}$$

Assume that the random effects covariance matrix $\mathbf{G}$ is positive definite and let $\lambda_{\max}$ and $\lambda_{\min}$ be its largest and smallest eigenvalue, respectively. Observe that $|\mathbf{\Sigma}| \to 0$, as $\lambda_{\max}$ approaches zero. This is the case when all the effects variance components are essentially zero and the mixed model reduces to a fixed effects model. Conversely, $|\mathbf{\Sigma}| \to |\mathbf{Z}'\bar{\mathbf{W}}\mathbf{Z}|^{-1}$, as $\lambda_{\min}$ diverges to infinity; that is, as the variability of the random effects increases without bound.

To illustrate, consider the OWMM with m groups and r observations per group. Let $\mathbf{G} = \sigma_1^2\mathbf{I}_m$, $\mathbf{W} = (1/\sigma_0^2)\mathbf{I}_n$, and $\mathbf{Z}_{n\times m} = \mathbf{I}_m \otimes \mathbf{1}_r$, where $n = m \times r$. It follows that

$$\mathbf{\Sigma} = [\mathbf{Z}'\mathbf{W}\mathbf{Z} + \mathbf{G}^{-1}]^{-1} = \left(\frac{r}{\sigma_0^2} + \frac{1}{\sigma_1^2}\right)^{-1}\mathbf{I}_m = c(\sigma_1^2)\mathbf{I}_m,$$

where $c(\sigma_1^2) = \sigma_0^2/(r + (\sigma_0^2/\sigma_1^2))$. With $\sigma_0^2(> 0)$ fixed, $c(\sigma_1^2)$ approaches 0 as $\sigma_1^2 \to 0$. On the other hand, as $\sigma_1^2 \to \infty$, $c(\sigma_1^2)$ converges to σ_0^2/r. Therefore, $|\mathbf{\Sigma}|$ is bounded between 0 and $(\sigma_0^2/r)^m$.

This implies, that regardless of the magnitude of the random effects variance components in $\mathbf{G}$, the generalized variance of Δ_{MCEM} has an upper bound, which is determined by the model and the data.

In contrast, we also show in the Appendix (Section A.5), that $|\boldsymbol{\tau}^2_{\text{SML}}| = O_p\left(|\mathbf{G}|^{1/2}\right)$. Hence $|\boldsymbol{\tau}^2_{\text{SML}}|$ is unbounded as a function of $\mathbf{G}$. Indeed, as the components of $\mathbf{G}$ can become arbitrarily large, the generalized variance of Δ_{SML} does not have an upper bound. In Section 5.2 we discuss two examples, both of which illustrate the superior efficiency of MCEM relative to SML when the effects variance is large.

5.1.2 Discussion

The generalized variances of Δ_{MCEM} and Δ_{SML} imply that, if the random effects variance components in $\mathbf{G}$ are small, both, MCEM and SML, will have small Monte Carlo error variances. But as the effects variance components approach zero,

the mixed model reduces to a fixed effects model. Thus, the interesting practical situations are when one or more of the effects variance components are large. Our results imply that the Monte Carlo variance of SML is unbounded relative to that of MCEM as the magnitude of the effects variance components increase. Therefore, when there is more variability in the data than can be accounted for by a fixed effects model, MCEM is generally a more precise and efficient method than SML.

A heuristic explanation of this result is that SML involves sampling from the (*marginal*) distribution of the random effects $\mathbf{u}$ which has variance $\text{Var}(\mathbf{u}) = \mathbf{G}$. If the sampling variance of a simulation-based estimation method is large, then parameter estimation is done in a population with large heterogeneity. Precise estimation in a highly heterogeneous population will in general be difficult. Indeed, as $\mathbf{G}$ is unbounded (component-wise), the sampling variance of SML can be arbitrarily large. This partially explains the erratic behavior of SML, which was also observed by McCulloch (1997).

Conversely, note that the sampling variance of MCEM is bounded. Indeed, in every iteration MCEM draws a random sample from the *conditional* distribution, $\mathbf{u}|\mathbf{y}$. Booth and Hobert (1998, Eq.12 & 14) derive $\text{Var}(\mathbf{u}|\mathbf{y}) \approx \boldsymbol{\Sigma}$, which is exact in the LMM. But as $\boldsymbol{\Sigma}$ is bounded (with respect to $\mathbf{G}$), MCEM samples from a population with finite and bounded variance, making a precise estimation more likely. This helps to explain the superior performance of MCEM.

Our derivations of the asymptotic variances $\boldsymbol{\tau}^2_{\text{MCEM}}$ and $\boldsymbol{\tau}^2_{\text{SML}}$ (given in the Appendix) involve Laplace approximations whose accuracy depends on both, sample size, n, and the effects variance, $\mathbf{G}$. However, our goal is to gain insight into the relative efficiencies of MCEM and SML *in terms of computational effort.* Both methods can, in principle, achieve any level of accuracy by simply increasing the Monte Carlo sample size M. In contrast, analytical approximations such as PQL (Breslow and Clayton, 1993) have a fixed level of accuracy determined by

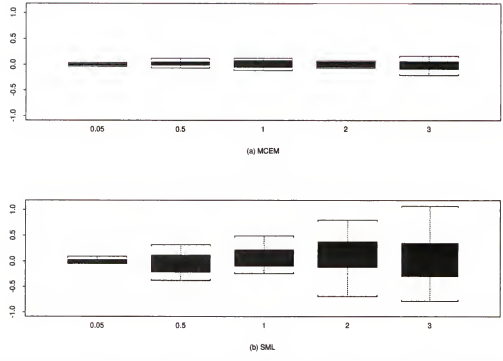

Figure 5.1: Variability of MCEM and SML for increasing variance components.

the model and the data. Since our derivations are exact for the LMM, one can expect the Laplace approximations to improve the more closely the GLMM can be approximated by a LMM. Finally, we note that our derivations assume the variance components are known. The examples in Section 5 indicate that our analytical results qualitatively predict the relative performance of the two algorithms in the unknown variance case also.

5.1.3 Simulation Study

Consider the OWMM for illustration. The OWMM is a very simple example of a GLMM, in which the MLE of μ is known, making the use of computationally intensive methods like MCEM and SML unnecessary. However, because of its simplicity, this model is very well suited to illustrate our results. Moreover, a

method failing in this simple model is unlikely to perform well in more complicated models.

Consider Figure 5.1. For 5 effects variance components, $\sigma_1^2 \in \{0.05; 0.5; 1; 2; 3\}$, 5 different sets of data were simulated from the OWMM (with fixed parameters $m = 10$, $r = 10$, $\mu = 0$ and $\sigma_0^2 = 1$). The parameter μ was estimated repeatedly in each data set, 50 times with MCEM (using 50 iterations and a constant Monte Carlo sample size of $M = 10$ per iteration) and 50 times with SML (using $M = 500$, making the total simulation amount of both methods equal). Box-plots of the estimates for μ are shown for (a) MCEM and (b) SML.

Figure 5.1 displays the variability of MCEM and SML for increasing values of σ_1^2. The variability of both methods is very small when σ_1^2 is almost zero ($\sigma_1^2 = 0.05$). This is the case where the mixed model essentially reduces to a fixed effects model and both methods perform very well.

On the other hand, as σ_1^2 increases, the variability of MCEM initially grows, but remains almost constant for $\sigma_1^2 \geq 1$. Conversely, the variability of SML continues to increase with every increase in σ_1^2 illustrating that the variability of SML is unbounded relative to that of MCEM.

5.2 Two Examples

The following two examples support the analytical results from Section 5.1. In addition, they also provide evidence that these results qualitatively extend to the more general case of estimating the complete parameter vector $\boldsymbol{\psi} = (\beta, \sigma^2)$.

The example in Section 5.2.1 is based on the Mississippi River data presented in Littell et al. (1996, Ch.4.2). In Section 5.2.2 we consider a data set generated according to a model that McCulloch (1997) used to illustrate MCEM and SML.

Table 5.1: Nitrogen Concentrations (in parts/million) at six randomly selected influents to the Mississippi River.

Influent1	Influent2	Influent3	Influent4	Influent5	Influent6
21	21	20	14	7	41
27	11	19	24	15	42
29	18	20	30	18	35
17	9	11	21	4	34
19	13	14	31	28	30
12	23		27		
29	2				
20					
20					

Table 5.2: Relative Efficiency of MCEM and SML for Mississippi data.

	Average		Empirical Variance	
	$\bar{\mu}$	$\bar{\sigma}_1^2$	s_μ^2	$s_{\sigma_1^2}^2$
MCEM	21.080	51.294	0.0051	0.2205
SML	21.225	51.624	1.6710	42.192
	Relative Efficiency		ρ 327.65	191.35

5.2.1 Mississippi River Data

Table 5.1 presents data on nitrogen concentrations from several sites at six randomly selected influents to the Mississippi River. Littell et al. (1996) propose an unbalanced version of the OWMM in equation (2.5) to fit this data. To be more specific, let r_i be the number of observations at influent i, $(i = 1, \ldots, 6)$, then they fit

$$y_{ij} = \mu + u_i + \epsilon_{ij},\ j = 1, \ldots, r_i;\ i = 1, \ldots, 6, \tag{5.8}$$

where the u_i's and ϵ_{ij}'s are random samples from $N(0, \sigma_1^2)$ and $N(0, \sigma_0^2)$, respectively. Littell et al. obtain the MLEs $\hat{\mu} = 21.22$, $\hat{\sigma}_0^2 = 42.7$ and $\hat{\sigma}_1^2 = 51.25$.

In order to measure the efficiency of MCEM and SML, we applied each method to the data 200 times, estimating μ and σ_1^2 (and treating $\hat{\sigma}_0^2$ as fixed

and known). In each replicate, MCEM was run for 20 iterations with a constant Monte Carlo sample size of $M = 1000$. 20 iterations were easily enough for the deterministic EM algorithm to converge. On the other hand, in each replication SML (using an importance sampling distribution with $\sigma_1^2 = \hat{\sigma}_1^2$) was applied with $M = 20,000$, making the total simulation amount for both methods equal. Table 5.2 presents the results. The average value of $\hat{\mu}$ over the 200 replications is denoted $\bar{\mu}$ and the variance of those 200 values of $\hat{\mu}$ is denoted s_{μ}^2. The corresponding quantities for $\hat{\sigma}_1^2$ are denoted by $\bar{\sigma}_1^2$ and $s_{\sigma_1^2}^2$.

Table 5.2 has some interesting features. Notice first, that for each of the two methods the variances for σ_1^2 are much larger than for μ. This indicates additional difficulties when estimating variance components. Furthermore, for each parameter we have computed the relative efficiency, ρ, defined as the ratio of the SML and MCEM variances. With the total simulation amount ($M = 20,000$) equal for both methods, the efficiency gain of MCEM ranges between $\rho = 327.65$ and $\rho = 191.35$ (for μ and σ_1^2, respectively). This means that MCEM is at least 191 times more efficient in using the total simulated data than SML.

5.2.2 McCulloch's Model

McCulloch (1997) considers a logistic-normal model, similar to the one described in Section 2.1.1. Specifically, suppose that the y_{ij} are conditionally independent with

$$y_{ij}|u_i \sim \text{Bernoulli}(\pi_{ij})$$

for $i = 1, ..., m$ and $j = 1, ..., r$, where

$$\eta_{ij} = \log\left(\frac{\pi_{ij}}{1 - \pi_{ij}}\right) = \beta x_{ij} + u_i.$$

The u_i are assumed to be a random sample from the $N(0, \sigma^2)$ distribution. McCulloch used data that were simulated according to this model with $m = 10$, $r = 15$,

$\beta = 5$, $\sigma^2 = .5$ and $x_{ij} = j/15$, but did not report the data. Booth and Hobert (1999) generated data, displayed in Table 5.3, using the same settings, and found the exact MLE to be $(\hat{\beta}, \hat{\sigma}^2) = (6.132, 1.766)$, based on numerical integration. We refer to this data set as the "original data". We also generated a second, new set of data, displayed in Table 5.4, with the same settings, which resulted in a MLE of $(\hat{\beta}, \hat{\sigma}^2) = (3.526, 0.270)$. We denote this data set as the "new data".

As in Section 5.2.1, for each data set we estimated β and σ^2 repeatedly (200 times) using each of the two methods, MCEM (20 iterations, $M = 1000$) and SML ($M = 20,000$), resulting in an equal total simulation amount for both methods.

Consider Tables 5.5 and 5.6. Notice, that for the original data the efficiency gain of MCEM is at least $\rho = 15.25$. However, for the new data, the two methods are almost equally efficient. This illustrates, that for a small variance component ($\hat{\sigma}^2 = 0.270$ for the new data), SML performs very well relative to MCEM. However, as the effects variance component increases, the efficiency of SML rapidly declines.

5.2.3 Practical Limitations of SML

In the context of these two examples, we want to point out some practical limitations and difficulties associated with SML. They apply, however, to all examples that we have considered in the framework of GLMMs.

We have found that SML is in practice hard to implement. One reason is that it requires a numerical method such as Newton-Raphson to find the maximum of the simulated likelihood function. However, numerical maximization routines (and in particular Newton-Raphson) often suffer from the need for good starting values. In our examples, we found it necessary to use starting values very close to the true MLEs. Using the same starting values as for MCEM, which were rather arbitrarily set to $(\mu^{(0)}, \sigma_1^{2(0)}) = (0, 1)$, always lead to convergence problems for SML. This is a serious disadvantage for SML, as it requires some prior knowledge about

Table 5.3: Original data according to McCulloch's model

i	Values for the following values of j														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	1	1	0	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
6	0	0	0	1	0	1	1	1	0	1	1	1	1	1	1
7	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 5.4: New data according to McCulloch's model

i	Values for the following values of j														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
3	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0
8	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1
9	0	0	1	1	0	0	0	1	1	1	1	0	1	1	1
10	1	0	0	0	1	1	1	0	1	1	1	1	1	1	1

Table 5.5: Relative Efficiency of MCEM and SML for McCulloch's model (1)

	Original Data $(\hat{\beta}, \hat{\sigma}^2) = (6.132, 1.766)$			
	Average		Empirical Variance	
	$\bar{\beta}$	$\bar{\sigma}^2$	s^2_β	$s^2_{\sigma^2}$
MCEM	6.1200	1.7531	0.0065	0.0100
SML	6.0975	1.6988	0.2325	0.1525
	Relative Efficiency	ρ	35.769	15.250

Table 5.6: Relative Efficiency of MCEM and SML for McCulloch's model (2)

	New Data $(\hat{\beta}, \hat{\sigma}^2) = (3.526, 0.270)$			
	Average		Empirical Variance	
	$\bar{\beta}$	$\bar{\sigma}^2$	s^2_β	$s^2_{\sigma^2}$
MCEM	3.5310	0.3094	0.0008	0.0008
SML	3.5267	0.2470	0.0021	0.0011
	Relative Efficiency	ρ	2.6250	1.3750

the solution at the outset. On the other hand, using "trial-and-error" to find good starting values can be very time consuming and is increasingly difficult in higher dimensional problems. Conversely, MCEM is very stable and generally converges even with starting values chosen randomly in the parameter space.

5.2.4 More Efficient Use of MCEM

We also want to emphasize that we have not used MCEM in its most efficient way in our examples. For simplification, we used MCEM with a *constant* Monte Carlo sample size M in each iteration. However, it is generally inefficient to use MCEM with a constant M in every iteration. Using smaller sample sizes in earlier iterations and increasing the sample sizes as the algorithm moves along leads to a more efficient use of the total simulation amount. This means that the Monte Carlo error in the last iteration will be significantly smaller than in our examples, making its superiority over SML even more apparent.

CHAPTER 6
EFFICIENCY IMPROVEMENT WITH QUASI-MONTE CARLO

So far we have considered stochastic estimation based on classical Monte Carlo methods; that is, based on methods that use random points to evaluate an intractable integral. In this chapter we investigate methods using *non*-random points. Specifically, we consider methods that are based on the ideas of Monte Carlo and which use deterministic sequences of points; in particular, sequences of points which are more uniformly spread in the sampling space than random points. The advantage of using these deterministic sequences is that the corresponding estimators often have a smaller variance which results in a more efficient use of the simulated data. Methods that are based on deterministic sequences are often classified as Quasi-Monte Carlo methods.

Quasi-Monte Carlo methods are relatively unpopular in statistics. Although there exist a variety of introductions and review articles of these methods (see, e.g., Shaw, 1988; Fang and Wang, 1994; Fang et al., 1994; Owen, 1998), the ideas of Quasi-Monte Carlo are not very often applied to practical statistical problems. Indeed, we found relatively few articles that make use of Quasi-Monte Carlo (Niederreiter and Peart, 1986; Tezuka and Fushimi, 1992; Carletti et al., 1994; Ostland and Yu, 1997; Pan and Thompson, 1998; Liao, 1998). This is somewhat surprising since this method has been used very successfully in other fields, for example in finance for the evaluation of high dimensional integrals occurring in financial derivatives (see Paskov and Traub, 1995; Paskov, 1997; Morokoff and Caflisch, 1998).

In Section 6.1 we introduce the basic concepts of Quasi-Monte Carlo, show how to construct deterministic sequences that are more uniformly spread than random points and point out advantages over classical Monte Carlo methods. In Section 6.2 we apply Quasi-Monte Carlo to the GHMs and investigate how Quasi-Monte Carlo can be used to improve the efficiency of SML.

6.1 Quasi-Monte Carlo Integration

Suppose we want to evaluate an (analytically intractable) integral

$$I = \int_{C^d} f(\mathbf{x}) d\mathbf{x} \tag{6.1}$$

over the d-dimensional *unit cube*, $C^d = [0,1]^d$. If d is small, then the integral in (6.1) is typically evaluated using standard (deterministic) quadrature methods; however, quadrature is no longer recommended when d is large. When d is large, (6.1) is often evaluated using Monte Carlo integration.

For a set of points, $\{\mathbf{x}_1, ..., \mathbf{x}_M\} \subset C^d$, classical Monte Carlo integration approximates (6.1) by the empirical average

$$\hat{I} = \frac{1}{M} \sum_{i=1}^{M} f(\mathbf{x}_i), \tag{6.2}$$

where the points $\mathbf{x}_i$ are selected *randomly* in the unit cube, that is, $\mathbf{x}_i \overset{iid}{\sim} \text{Unif}([0,1]^d)$.

One criticism of classical Monte Carlo integration is that its ideas are based on the ability to produce random points $\mathbf{x}_i$. However, in reality random points are typically not available. In fact, in most cases, the points $\mathbf{x}_i$ are produced with the help of a random number generator; but since random number generators are based

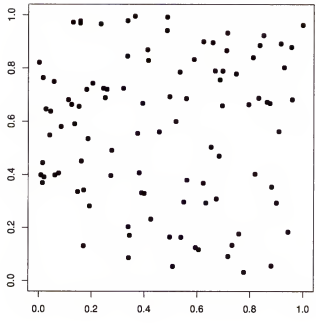

Figure 6.1: 100 random points in the unit square.

on deterministic algorithms [1] , these points cannot be considered genuinely random and are therefore often referred to as "pseudo"-random points.

Another criticism is that random numbers typically do not explore the sample space very well. Consider Figure 6.1, for instance, which shows 100 random points of the unit square, $\mathbf{x}_i \overset{iid}{\sim} \text{Unif}([0,1]^2), i = 1, \ldots, 100$. Random points tend to form clusters, "over-sampling" the unit square in some places; this leads to gaps in other places, where the sample space is not explored at all.

Quasi-Monte Carlo methods avoid these points of criticism. Although the basic ideas of Monte Carlo and Quasi-Monte Carlo integration are very similar, there is one important difference: instead of using random points, Quasi-Monte Carlo

[1] See, for example, Robert and Casella (1999, Chapter 2) for a more thorough discussion of this issue.

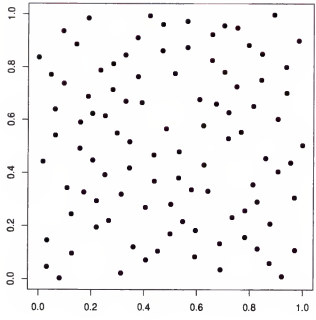

Figure 6.2: 100 points of a low discrepancy sequence in the unit square.

uses a deterministic sequence of points which explores the sample space better than random points. These sequences are often referred to as *low discrepancy* sequences.

6.1.1 Low Discrepancy Sequences

Quasi-Monte Carlo uses deterministic sequences of points which provide a better spread, or "uniformity", in the sampling space, avoiding the gaps and clusters that arise from random sampling. In order to quantify the notion of uniformity, one defines a distance measure, the *discrepancy.* Several different distance measures exist; the most widely studied measure is the *star discrepancy* which is defined as follows (see, e.g., Fang et al., 1994; Morokoff and Caflisch, 1995).

Let R be a rectangle contained in C^d with sides parallel to the coordinate axes, and let $\lambda(R)$ denote the Lebesgue measure of R. The star discrepancy, D_M^*, for the

sequence $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ is

$$D_M^* = \sup_{R \subset C^d} \left| \frac{\#\text{ of points } \mathbf{x}_i \text{ in } R}{M} - \lambda(R) \right|. \tag{6.3}$$

Quasi-Monte Carlo methods use sequences of points with the smallest possible discrepancy. One can show that for a set of random points the star discrepancy has asymptotic order of $O_P((\log\log M)/M^{1/2})$. On the other hand, there exist many deterministic sequences that have smaller asymptotic order. The star discrepancy of the best-known deterministic sequences have asymptotic order of $O((\log M)^d/M)$; these sequences are called low discrepancy sequences. The asymptotic order suggests that low discrepancy sequences have greater uniformity than random sequences. Notice, however, that for large d it could take (impractically) large sample sizes, M, before the asymptotics are relevant. However, empirical studies suggest that Quasi-Monte Carlo can be more accurate than Monte Carlo on some real problems with practical sample sizes (see, e.g., Morokoff and Caflisch, 1995).

Figure 6.2 shows 100 points of a low discrepancy sequence in the unit square. Compared to Figure 6.1, these points avoid clustering and provide a more uniform spread than random points.

There exist many different low discrepancy sequences. Examples include the Halton sequence (Halton, 1960), the Sobol sequence (Sobol, 1967), the Faure sequence (Faure, 1982), and the Niederreiter sequence (Niederreiter, 1992). In this work we will focus on the Halton sequence only.

6.1.2 Halton Sequences

Let b, $b \geq 2$, be an integer. Then any integer n, $n \geq 0$, can be written in base-b representation as

$$n = d_j b^j + d_{j-1} b^{j-1} + \cdots + d_1 b + d_0,$$

where $d_i \in \{0, 1, \ldots, b-1\}$ for $i = 0, 1, \ldots, j$. For example, if $b = 7$, we can write the integer $n = 101$ as $n = 2 \cdot 7^2 + 3 \cdot 7^0$.

The *radical inverse function* $\phi_b(n)$ of n to the base b is defined as

$$\phi_b(n) = \frac{d_0}{b^1} + \frac{d_1}{b^2} + \ldots + \frac{d_j}{b^{j+1}}.$$

For example, for the base $b = 7$ and $n = 101$, the radical inverse function applied to n gives $\phi_7(101) = 3/7 + 2/7^3$. Notice that $\phi_b(\cdot)$ maps every integer onto an element of the unit interval; that is, for every integer, $n \geq 0$, $\phi_b(n) \in [0, 1]$.

The radical inverse function defines the elements of a Halton sequence. If we let $b_1, ..., b_d$ be d different prime integers that are greater than one, then a d-dimensional Halton sequence is given by $\{\mathbf{x}_0, ..., \mathbf{x}_{M-1}\} \subset C^d$, where the elements $\mathbf{x}_k$ are defined by

$$\mathbf{x}_k = (\phi_{b_1}(k), ..., \phi_{b_d}(k))', k = 0, 1, ..., M-1. \tag{6.4}$$

In practice, the integers $b_1, ..., b_d$ are often chosen to be the first d prime numbers.

Notice that we do not have to start the sequence (6.4) at the origin, $k = 0$. For any d-vector of integers, say $\mathbf{m} = (m_1, ..., m_d)', m_i \geq 0$, the sequence defined by

$$\mathbf{x}_k = (\phi_{b_1}(m_1 + k), ..., \phi_{b_d}(m_d + k))', k = 0, 1, ..., M-1 \tag{6.5}$$

is still a low discrepancy sequence (see, e.g., Pagès, 1992; Bouleau and Lépingle, 1994).

6.1.3 Approximation Error of Quasi-Monte Carlo

One disadvantage of Quasi-Monte Carlo methods is that estimation of the approximation error is typically very complicated. An error bound for approximations of the form (6.2) is given by the Koksma-Hlawka inequality (see, e.g., Fang and

Wang, 1994),

$$\left|I - \hat{I}\right| \leq V(f) D_M^*, \tag{6.6}$$

where $V(f)$ denotes the total variation of f in the sense of Hardy and Krause (see Niederreiter, 1978, p.966). The error bound in (6.6) has only limited usefulness, since $V(f)$ is difficult to estimate. Classical Monte Carlo methods typically estimate the error in (6.6) by techniques based on the Strong Law of Large Numbers and the Central Limit Theorem. These techniques, however, require the sequence of points, $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$, to be random. Since Quasi-Monte Carlo methods are based on deterministic sequences, these techniques do not apply. This drawback has lead to the development of *randomized* Quasi-Monte Carlo integration.

6.1.4 Randomized Quasi-Monte Carlo

Several authors have suggested using randomized low discrepancy sequences (Shaw, 1988; Owen, 1998; Wang and Hickernell, 2001). Owen (1998) proposes that a randomized low discrepancy sequence, $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, should have the following properties:

1. Every element of the sequence is a random point of the unit cube; that is, $\mathbf{x}_i \sim \text{Unif}([0,1]^d), \forall i$.
2. The sequence $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ is a low discrepancy sequence with probability one.

Property 1 makes the estimator in equation (6.2) an unbiased estimate of I and property 2 preserves the uniformity of the sequence of points.

For R independent randomized low discrepancy sequences let $\hat{I}^{(1)}, \ldots, \hat{I}^{(R)}$ be the corresponding estimates based on equation (6.2). The variance of the (randomized) Quasi-Monte Carlo estimate can then be estimated by

$$\frac{1}{R(R-1)} \sum_{j=1}^{R} (\hat{I}^{(j)} - \hat{I})^2, \tag{6.7}$$

where $\hat{I} = \sum_{j=1}^{R} \hat{I}^{(j)}/R$. One can thus use equation (6.7) to estimate the approximation error in (6.6).

6.1.5 Randomized Halton Sequences

We mentioned in Section 6.1.2 that the Halton sequence in (6.5) is a low discrepancy sequence regardless of the starting point. Wang and Hickernell (2001) use this fact to develop Halton sequences with random starting points.

For simplicity, let us consider the univariate case only, $d = 1$. The extension to the multivariate case is straightforward. Let x_0 be a randomly chosen point in the unit interval, $x_0 \sim \text{Unif}([0, 1])$, and let b be a prime integer. Since the construction of the Halton sequence is based on the radical inverse function, which operates on the integers, we need to find a suitable integer-representation for x_0. Notice that we can write x_0 in base-b notation as

$$x_0 = \sum_{j=0}^{\infty} \frac{d_j}{b^{j+1}} \approx \sum_{j=0}^{J} \frac{d_j}{b^{j+1}} = \tilde{x}_0,$$

say, for $d_j \in \{0, 1, ..., b-1\}$. The integer J is typically a large number and chosen to satisfy a predefined accuracy (which could be, for example, higher than the actual computer accuracy). The base-b integer-representation of the starting point $\tilde{x}_0$ is then

$$m \equiv m(\tilde{x}_0) = d_J b^J + ... + d_0 b^0 = \phi_b^{-1}(\tilde{x}_0).$$

Therefore, since x_0 is random, so is the integer m. The randomized (univariate) Halton sequence, $\{x_0, ..., x_{M-1}\}$, starts at the (random) integer m and computes successive elements according to

$$x_k = \phi_b(m + k), k = 0, \ldots, M - 1. \tag{6.8}$$

Wang and Hickernell (2001) show that if the starting point of the Halton sequence is random, then every consecutive point of the sequence is random also,

$$x_0 \sim \text{Unif}([0,1]) \Longrightarrow \phi_b(m(x_0)+k) \sim \text{Unif}([0,1]), k = 1,2,3,\ldots . \tag{6.9}$$

Therefore, the randomized Halton sequence satisfies properties 1 and 2 of Section 6.1.4.

6.2 Quasi-Monte Carlo methods in GLMMs

To this point we have introduced Quasi-Monte Carlo methods and outlined how to approximate integrals over the *unit cube* with the help of these methods. In this section we show how Quasi-Monte Carlo can be used to approximate the intractable likelihood function in (2.6). Recall that the likelihood involves an integral of the form

$$L(\boldsymbol{\psi}|\mathbf{y}) = \int_{\mathcal{R}} f(\mathbf{y}|\mathbf{u};\boldsymbol{\psi}_1) g(\mathbf{u};\boldsymbol{\psi}_2) d\mathbf{u}, \tag{6.10}$$

where $\mathcal{R}$ denotes the range of integration. Except for few special cases, the range of integration in (6.10) will be different from the unit cube, C^d. Therefore, Quasi-Monte Carlo methods can typically not be applied directly to the integral in (6.10).

In order to apply Quasi-Monte Carlo to (6.10) one has to find a suitable transformation from $\mathcal{R}$ into the unit cube. Clearly, such a transformation will only exist for few special cases of the GHM. The GLMM, by its very nature, is an example of a GHM for which such a transformation often exists.

6.2.1 Probability Integral Transformation for GLMMs

In a GLMM one typically assumes normality for the random effects, that is, one assumes that $g(\mathbf{u};\boldsymbol{\psi}_2) \sim N(\mathbf{0},\mathbf{G})$. If we assume that, in addition, the random effects are independent, that is, if we assume that $\mathbf{G}$ is a diagonal matrix with

diagonal elements $\sigma_i^2, i = 1, \ldots, q$, then we can write equation (6.10) as

$$L(\boldsymbol{\psi}|\mathbf{y}) = \int f(\mathbf{y}|u_1, \ldots, u_q; \boldsymbol{\psi}_1) \prod_{i=1}^{q} \frac{\varphi(u_i/\sigma_i)}{\sigma_i} du_1, \ldots, du_q, \tag{6.11}$$

where φ denotes the pdf of the standard normal distribution. Using the *probability integral transformation*, $v_i = \Phi(u_i/\sigma_i)$, where $\Phi(x)$ denotes the cdf of the standard normal distribution, equation (6.11) becomes

$$L(\boldsymbol{\psi}|\mathbf{y}) = \int_{C^q} f(\mathbf{y}|\Phi^{-1}(\sigma_1 v_1), \ldots, \Phi^{-1}(\sigma_q v_q); \boldsymbol{\psi}_1)\, dv_1, \ldots, dv_q. \tag{6.12}$$

Equation (6.12) makes the application of Quasi-Monte Carlo straightforward.

We point out that only few practical problems will allow a representation of the form (6.12). For instance, if the dependence structure of the random effects is more complex (than the assumed independence above), then this method will not work. Although there exist further useful transformation methods (see Robert and Casella, 1999, Chapter 2.2), the practitioner may often encounter situations where a transformation into the unit cube does not exist.

6.2.2 Efficiency of SML using Quasi-Monte Carlo

In the following two examples we investigate how the efficiency of SML can be improved by using Quasi-Monte Carlo sampling instead of classical Monte Carlo.

OWMM. Consider the OWMM introduced in Section 2.1.3 and assume that the variance components, σ_0^2 and σ_1^2, are known and that we are only interested in estimating the general mean μ.

For each of the following eight effects variance components, $\sigma_1^2 \in \{0.05; 0.1; 0.2; 0.4; 0.6; 0.8; 1; 1.5\}$, we simulated 20 sets of data from the OWMM (with μ and σ_0^2 held fixed at 0 and 1, respectively). For each set of data, we estimated μ 50 times using SML with each of the two sampling methods (and fixed Monte Carlo sample size M). For each value of σ_1^2, we computed the mean squared errors for the estimates based on classical Monte Carlo (say, $\widehat{\text{MSE}}_{\text{MC}}$) and randomized Quasi-Monte

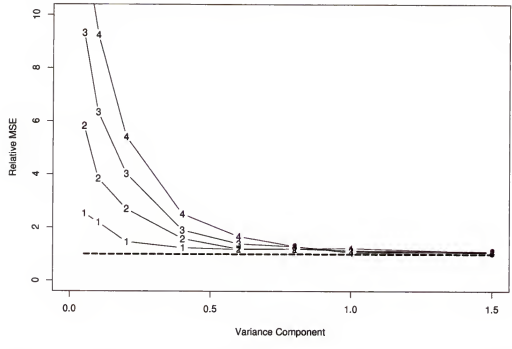

Figure 6.3: Relative Mean Squared Errors of Quasi-Monte Carlo for OWMM.

Carlo (say, $\widehat{\text{MSE}}_{\text{QMC}}$). We repeated the experiment for four different values of M, $M \in \{10; 50; 100; 200\}$. Figure 6.3 shows plots of the *relative* mean squared errors, $\widehat{\text{MSE}}_{\text{MC}}/\widehat{\text{MSE}}_{\text{QMC}}$, as a function of σ_1^2 for $M = 10$ (line 1), $M = 50$ (line 2), $M = 100$ (line 3) and $M = 200$ (line 4). The dotted line in Figure 6.3 represents equal mean squared errors, that is, $\widehat{\text{MSE}}_{\text{MC}} = \widehat{\text{MSE}}_{\text{QMC}}$.

Observe that the relative mean squared errors increase as σ_1^2 decreases; that is, the smaller σ_1^2, the higher the gain from using randomized Quasi-Monte Carlo over classical Monte Carlo. Moreover, this gain becomes larger for larger Monte Carlo sample sizes. This implies that the advantage from using more uniformly distributed low discrepancy sequences increases the more points we sample. Classical Monte Carlo performs almost as well when we sample only few points. However, with increasing M, Quasi-Monte Carlo methods gain over classical Monte Carlo, since they avoid clusters and explore the sample space more thoroughly.

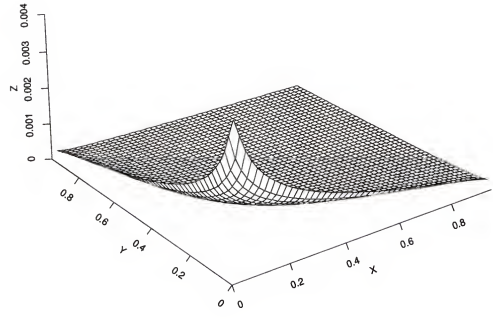

Figure 6.4: Integrand of likelihood in (6.12) for $\sigma_1^2 = 0.1$

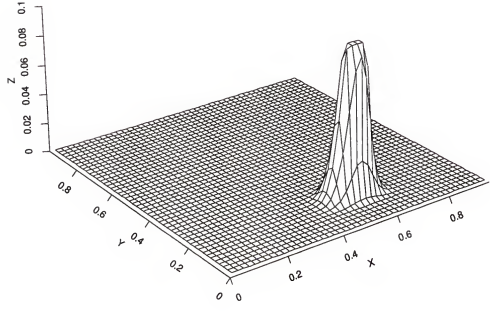

Figure 6.5: Integrand of likelihood in (6.12) for $\sigma_1^2 = 2$

Another interesting feature in Figure 6.3 is that the gain in using Quasi-Monte Carlo decreases as the effects variance component σ_1^2 increases. Morokoff and Caflisch (1995, p.218) found in empirical studies that "Quasi-Monte Carlo is superior to classical Monte Carlo, but the advantage may be slight [...] for integrands that are not smooth". Figure 6.4 shows the plot of the likelihood-integrand in (6.12) for a two-dimensional random effect, $\mathbf{u} \in \mathcal{R}^2$, with $\sigma_1^2 = 0.1$. We can see that the integrand is a very smooth curve in the unit square. Conversely, Figure 6.5 shows a plot of the integrand for a large variance component ($\sigma_1^2 = 2$). Clearly, for this variance component the integrand is very non-smooth and spiky.

Logistic-Normal Model. For the logistic-normal model described in Section 5.2.2 (using the "original" data with MLEs $\hat{\beta} = 6.132$ and $\hat{\sigma}^2 = 1.766$) we conducted a simulation study similar to the previous example. We estimated both parameters repeatedly, 100 times using SML with classical Monte Carlo and 100 times using SML with randomized Quasi-Monte Carlo. We performed the experiment two times, with $M = 1,000$ in the first run and $M = 5,000$ in the second run. Table 6.1 shows the bias and mean squared error for the estimates based on Monte Carlo sampling (MC) as well as randomized Quasi-Monte Carlo (QMC) and their ratios (MC/QMC).

We can see that SML based on randomized Quasi-Monte Carlo outperforms classical Monte Carlo in terms of the bias as well as the mean squared error for both parameter components. Although the efficiency gain may be small, it increases as the Monte Carlo sample size increases. Since the variance component, $\hat{\sigma}^2 = 1.766$, can be considered rather large (implying a potentially very non-smooth likelihood integrand), we could expect an even larger gain for smaller values of $\hat{\sigma}^2$.

6.3 Conclusion

Quasi-Monte Carlo is an appealing alternative to classical Monte Carlo. The concept of deterministic sequences that explore the sampling space more

Table 6.1: Relative Efficiency of Quasi-Monte Carlo for McCulloch's model

	Original Data $(\hat{\beta}, \hat{\sigma}^2) = (6.132, 1.766)$					
	β			σ^2		
	MC	QMC	MC/QMC	MC	QMC	MC/QMC
Bias	0.01236	0.01205	1.02573	-0.13340	-0.04561	2.92480
MSE	0.14425	0.10013	1.44063	0.43418	0.27155	1.59890
	(Monte Carlo Sample Size $M = 1000$)					
Bias	0.01694	0.00113	14.9912	-0.03817	-0.00627	6.08772
MSE	0.05802	0.02062	2.81377	0.33499	0.14184	2.36174
	(Monte Carlo Sample Size $M = 5000$)					

thoroughly than random points has great theoretical appeal and has certainly also some practical value as pointed out in the previous examples.

However, application of Quasi-Monte Carlo to GHMs is limited. The need for a suitable transformation into the unit cube makes Quasi-Monte Carlo applicable only in few special cases. For SML, when the distribution of the random effects is normal (or also for other "standard" distributions, like exponential, Poisson, beta or gamma), such a transformation may often be available. However, for methods like MCEM, that require sampling from an often very complicated, conditional distribution, this may not be the case.

Moreover, the OWMM example has shown that the gain in using Quasi-Monte Carlo integration may be very small for situations with very non-smooth integrands, like the SML-integrand for large effects variance components. However, in the context of mixed models situations with larger variance components are certainly of greater interest, since, if the effects variance components are small, the mixed model can often be approximated sufficiently well by a fixed effects model (for which Monte Carlo methods are not needed at all).

CHAPTER 7
EFFICIENCY OF STOCHASTIC APPROXIMATION

The previous two chapters focused on investigating the performance of SML relative to that of MCEM. Now we turn our attention to stochastic approximation procedures. As we have already pointed out, stochastic approximation methods have an apparent advantage over methods like MCEM, since they reduce the error variance without increasing the Monte Carlo sample size. In the following we compare the performance of SAEM with that of MCEM.

7.1 Convergence Rate of MCEM and SAEM

Stochastic algorithms, like MCEM or SAEM, can be decomposed into two components; a deterministic component, that follows the underlying deterministic version of the algorithm, and a random noise component due to Monte Carlo error. Just as the deterministic version of MCEM is the EM algorithm, there is also a deterministic version underlying SAEM. Replacing $\tilde{Q}$ by Q (or $\tilde{\mathbf{F}}$ by $\mathbf{F}$) in equation (4.32) (or (4.33)), reveals the underlying deterministic version of SAEM.

In order to investigate the convergence behavior of a stochastic algorithm, both of these components have to be examined. The noise component determines the variability of the parameter update about the "average" update, the update of the underlying deterministic version. This variability can, at least in principle, be made arbitrarily small by simply increasing M, the Monte Carlo sample size. In the limit, as M approaches infinity, all noise is removed from the system and the rate at which the algorithm converges is entirely determined by its deterministic component. In the following we investigate the convergence rates of the deterministic components of MCEM and SAEM.

Our starting point is the approximate relation in (3.17); that is, in a neighborhood of the MLE the EM update satisfies

$$\boldsymbol{\psi}^{(t+1)} - \hat{\boldsymbol{\psi}} \approx \mathbf{B}(\boldsymbol{\psi}^{(t)} - \hat{\boldsymbol{\psi}}), \tag{7.1}$$

where $\boldsymbol{\psi}$ $(S \times 1)$ and $\mathbf{B}$ $(S \times S)$ is the rate matrix defined in equation (3.18). Equation (7.1) implies that the rate at which EM converges is $\mathbf{B}^{t+1}$, since $\boldsymbol{\psi}^{(t+1)} - \hat{\boldsymbol{\psi}} \approx \mathbf{B}^{t+1}(\boldsymbol{\psi}^{(0)} - \hat{\boldsymbol{\psi}})$.

Often, it is desirable to have a univariate measure for the convergence rate. Let $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_S)'$ denote the parameter components and $b_1, \ldots, b_S$ the eigenvalues of $\mathbf{B}$. It follows from the Spectral Decomposition Theorem and equation (7.1) that if ψ_j, $j = 1, \ldots, S$, is associated with a *large* b_j, then this parameter component converges at a *slow* rate (and vice versa). Therefore, some authors suggest using b_{max}, the largest eigenvalue of $\mathbf{B}$, as a measure of the convergence rate of EM (see Dempster et al., 1977; Laird et al., 1987; Meng, 1994). Clearly, b_{max} measures the convergence of the slowest component of $\boldsymbol{\psi}$. However, it does not provide an overall measure of the convergence rate.

On the other hand, recall that the determinant of $\mathbf{B}$ can be expressed as the product of its eigenvalues,

$$|\mathbf{B}| = \prod_{j=1}^{S} b_j. \tag{7.2}$$

Since $|\mathbf{B}|$ is a monotonically increasing symmetric function of the eigenvalues, it provides a good measure for the overall convergence rate of an algorithm [1] . Thus,

[1] In similar fashion, in multivariate analysis where $\mathbf{S}$ denotes sample covariance matrix, one often uses $|\mathbf{S}|$ rather than the largest eigenvalues of $\mathbf{S}$ as a univariate measure of the overall scatter about the mean (see Mardia et al., 1979).

EM's overall convergence rate can be expressed as

$$|\mathbf{B}|^{t+1} = \prod_{j=1}^{S} b_j^{t+1}. \tag{7.3}$$

Next we establish that all eigenvalues of $\mathbf{B}$ lie in the interval $[0, 1]$. Let $\mathbf{J}_o \equiv \mathbf{J}(\hat{\psi})$ be the *observed-data* information matrix, where $\mathbf{J}(\psi)$ is the negative second derivative of the log-likelihood of ψ,

$$\mathbf{J}(\psi) = -\frac{\partial^2 \log L(\psi|\mathbf{y})}{\partial\psi\partial\psi'}, \tag{7.4}$$

with $L(\psi|\mathbf{y})$ defined in (2.6). On the other hand, the *complete-data* information matrix is given by

$$-\left.\frac{\partial^2 \log f(\mathbf{y}, \mathbf{u}; \psi)}{\partial\psi\partial\psi'}\right|_{\psi=\hat{\psi}}. \tag{7.5}$$

Since $\mathbf{u}$ is unobserved, one averages (7.5) over the conditional distribution of $\mathbf{u}$ given $\mathbf{y}$ to get

$$\begin{aligned} \mathbf{J}_c &\equiv -E\left[\left.\frac{\partial^2 \log f(\mathbf{y}, \mathbf{u}; \psi)}{\partial\psi\partial\psi'}\right| \mathbf{y}; \psi\right]\Bigg|_{\psi=\hat{\psi}} \\ &= -\left.\frac{\partial^2 Q(\psi_*|\psi)}{\partial\psi_*\partial\psi_*'}\right|_{\psi_*,\psi=\hat{\psi}}. \end{aligned} \tag{7.6}$$

From the factorization

$$f(\mathbf{y}, \mathbf{u}; \psi) = L(\psi|\mathbf{y}) f(\mathbf{u}|\mathbf{y}; \psi), \tag{7.7}$$

it follows that the log-likelihood of ψ is

$$\log L(\psi|\mathbf{y}) = \log f(\mathbf{y}, \mathbf{u}; \psi) - \log f(\mathbf{u}|\mathbf{y}; \psi). \tag{7.8}$$

Equation (7.8) implies that after taking second derivatives, averaging with respect to $f(\mathbf{u}|\mathbf{y}; \psi)$, and evaluating at $\psi = \hat{\psi}$, we get the fundamental identity,

$$\mathbf{J}_o = \mathbf{J}_c - \mathbf{J}_m, \tag{7.9}$$

which is often found useful in the context of the EM algorithm. In equation (7.9) we defined the matrix

$$\mathbf{J}_m \equiv -E\left[\frac{\partial^2 \log f(\mathbf{u}|\mathbf{y};\boldsymbol{\psi})}{\partial\boldsymbol{\psi}\partial\boldsymbol{\psi}'}\middle|\mathbf{y};\boldsymbol{\psi}\right]\Bigg|_{\boldsymbol{\psi}=\hat{\boldsymbol{\psi}}}, \tag{7.10}$$

which can be viewed as the missing information. Oakes (1999) shows that

$$\mathbf{J}_m = \frac{\partial^2 Q(\boldsymbol{\psi}_*|\boldsymbol{\psi})}{\partial\boldsymbol{\psi}_*\partial\boldsymbol{\psi}'}\Bigg|_{\boldsymbol{\psi}_*,\boldsymbol{\psi}=\hat{\boldsymbol{\psi}}}. \tag{7.11}$$

Equation (7.9) can be written in the appealing form

$$\text{observed information} = \text{complete information} - \text{missing information},$$

which is often referred to as the missing information principle (see, e.g., Louis, 1982; Meng and Rubin, 1991).

Dempster et al. (1977) as well as Meng and Rubin (1991). show that for the rate matrix

$$\mathbf{B} = \mathbf{J}_m\mathbf{J}_c^{-1} \tag{7.12}$$

(see, e.g., Dempster et al., 1977; Meng and Rubin, 1991). It follows from equations (7.9) and (7.12) that

$$\mathbf{J}_o = (\mathbf{I} - \mathbf{B})\mathbf{J}_c. \tag{7.13}$$

Notice that if $\hat{\boldsymbol{\psi}}$ is a local, if not global, maximum of $\log L(\boldsymbol{\psi}|\mathbf{y})$, then, necessarily, $\mathbf{J}_o$ is positive (semi-) definite. Furthermore, Dempster et al. (1977, Theorem 4) show that $\mathbf{J}_c$ is positive definite. It follows (see Mardia et al., 1979, Corollary A.7.3.1) that all non-zero eigenvalues of $\mathbf{I}-\mathbf{B}$ are positive; that is, $1-b_j \geq 0$, where b_j denotes the jth eigenvalue of $\mathbf{B}$. This implies that all eigenvalues of $\mathbf{B}$ are real and less than (or equal to) one. Furthermore, since

$$\mathbf{J}_m = \text{Var}\left(\frac{\partial \log f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})}{\partial\boldsymbol{\psi}}\middle|\mathbf{y};\boldsymbol{\psi}\right)\Bigg|_{\boldsymbol{\psi}=\hat{\boldsymbol{\psi}}} \tag{7.14}$$

(see Louis, 1982), $\mathbf{J}_m$ is also positive (semi-) definite. It follows from equation (7.12) that all non-zero eigenvalues of $\mathbf{B}$ are positive. In summary, all eigenvalues of $\mathbf{B}$ lie in the interval $[0, 1]$.

Consider the univariate case for an intuitive explanation of this result. Let $\psi \in \mathcal{R}^1$ and let b be the rate constant satisfying the approximate relation in (7.1). Then, equation (7.12) implies that

$$\text{missing information} = b \times \text{complete information}.$$

Since the missing information cannot be negative and since it cannot exceed the complete information, necessarily $b \in [0, 1]$.

We now derive the deterministic component of the SAEM algorithm. Recall that the $(t+1)$st SAEM update satisfies $\bar{\mathbf{F}}_{t+1}(\boldsymbol{\psi}) = \mathbf{0}$, with $\bar{\mathbf{F}}_{t+1}$ defined in (4.33). Consequently, replacing $\bar{\mathbf{F}}$ by $\mathbf{F}$ in (4.33) yields the estimating equations for the deterministic component of SAEM; that is, the $(t+1)$st update of the deterministic SAEM algorithm solves $\mathbf{G}_{t+1}(\boldsymbol{\psi}) = \mathbf{0}$ where

$$\mathbf{G}_{t+1}(\boldsymbol{\psi}) = (1 - \gamma_t)\mathbf{G}_t(\boldsymbol{\psi}) + \gamma_t \mathbf{F}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(t)}). \tag{7.15}$$

An iterative argument shows that

$$\mathbf{G}_{t+1}(\boldsymbol{\psi}) = \sum_{i=0}^{t} w_{t,i} \mathbf{F}(\boldsymbol{\psi}, \boldsymbol{\psi}^{(i)}), \tag{7.16}$$

where $w_{t,i} = \gamma_i \prod_{j=i+1}^{t}(1 - \gamma_j)$ and $\sum_{i=0}^{t} w_{t,i} = 1$ (by induction). Using EM's approximate relation in equation (7.1), the $(t+1)$st update of the deterministic SAEM algorithm can then be written as

$$\boldsymbol{\psi}^{(t+1)} - \hat{\boldsymbol{\psi}} \approx \sum_{i=0}^{t} w_{t,i} \mathbf{B}(\boldsymbol{\psi}^{(i)} - \hat{\boldsymbol{\psi}}). \tag{7.17}$$

A standard weighting scheme is (see, e.g., McCulloch and Searle, 2001)

$$\gamma_t = \frac{B}{A+t}, \; t = 0, 1, 2, \ldots, \tag{7.18}$$

where A and B are positive integers. Taking $A = B = 1$, it follows that $w_{t,i} = 1/(t+1)$. Straightforward calculations show that equation (7.17) then reduces to

$$\psi^{(t+1)} - \hat{\psi} \approx \bar{\mathbf{B}}_{t+1}(\psi^{(0)} - \hat{\psi}), \tag{7.19}$$

where the convergence rate of deterministic SAEM is determined by the matrix

$$\bar{\mathbf{B}}_{t+1} = \prod_{k=0}^{t} \frac{\mathbf{B} + k\mathbf{I}}{1+k}. \tag{7.20}$$

In the following we compare the convergence rates of EM and deterministic SAEM. We start our discussion with the univariate parameter case.

7.1.1 Univariate Parameter Case

Assume that $\psi \in \mathcal{R}^1$ and let the rate constant b satisfy (7.1). It follows that EM's convergence rate is b^{t+1}. On the other hand, the scalar version of the matrix in equation (7.20) is

$$\bar{b}_{t+1} \equiv \prod_{k=0}^{t} \frac{b+k}{1+k} = b\frac{b+1}{2}\frac{b+2}{3}\cdots\frac{b+t}{t+1} = \frac{\Gamma(b+t+1)}{\Gamma(b)\Gamma(t+2)} \tag{7.21}$$

If we let $f(t) \sim g(t)$ denote asymptotic equivalence of f and g as $t \to \infty$, then the convergence rate of deterministic SAEM is

$$\bar{b}_{t+1} \sim \frac{1}{\Gamma(b)}\, t^{b-1}, \tag{7.22}$$

which follows from Abramowitz and Stegun (1992, p.257). Thus, the deterministic algorithm underlying SAEM converges at an algebraic rate, t^{b-1}, compared to the exponential rate, b^t, of EM. That is, EM converges at a much faster rate than

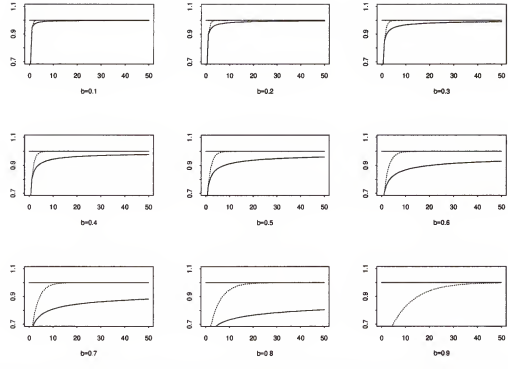

Figure 7.1: Convergence of EM and deterministic SAEM for different values of b.

deterministic SAEM. The following examples illustrate the performance of both deterministic algorithms for different values of b.

Figure 7.1 shows the convergence rate of EM (dashed lines) and deterministic SAEM (solid lines) for the values b between 0.1 and 0.9. Clearly, EM outperforms deterministic SAEM. In particular, SAEM's convergence rate is decent relative to that of EM for values of b close to 0. However, it deteriorates fast as b approaches one.

7.1.2 Multivariate Parameter Case

We now focus our attention on the multivariate parameter case, $\boldsymbol{\psi} \in \mathcal{R}^S$. The following theorem generalizes the results from Section 7.1.1.

Theorem 7.1. *Suppose the rate matrix* $\mathbf{B}$ *in equation (3.18) has* S *eigenvalues,* $0 \leq b_j \leq 1, j = 1, \ldots, S$. *Let* t *denote the iteration number. Then,*

(i) MCEM converges at a rate of $|\mathbf{B}|^{t+1}$*;*

(ii) SAEM, with weights of the form $\gamma_t = 1/(1+t)$*, converges at a rate of* $|\mathbf{B}|\ t^{tr(\mathbf{B})-S}/c$*, where* $c \equiv \prod_{j=1}^{S} \Gamma(b_j+1)$.

Proof. We have already shown that EM's convergence rate in the multivariate case is $|\mathbf{B}|^{t+1}$. To establish (ii), we define

$$\mathcal{C}_{t+1} = \frac{\mathbf{B}}{t+1} \text{ and } \mathcal{D}_t = \prod_{k=1}^{t}(\mathbf{I}+\mathbf{B}/k). \tag{7.23}$$

Then we can write equation (7.20) as $\bar{\mathbf{B}}_{t+1} = \mathcal{C}_{t+1}\mathcal{D}_t$. Notice that

$$|\mathcal{D}_t| = \prod_{k=1}^{t}|\mathbf{I}+\mathbf{B}/k| = \prod_{k=1}^{t}\prod_{j=1}^{S}(1+\frac{b_j}{k}). \tag{7.24}$$

We have already argued that

$$\prod_{k=1}^{t}(1+\frac{b_j}{k}) \sim \frac{t^{b_j}}{\Gamma(b_j+1)} \tag{7.25}$$

Equations (7.24) and (7.25) imply that

$$|\mathcal{D}_t| = t^{\sum_{j=1}^{S} b_j}/\prod_{j=1}^{S}\Gamma(b_j+1) = t^{tr(\mathbf{B})}/c. \tag{7.26}$$

Part (ii) of the theorem now follows, since $|\mathcal{C}_{t+1}| = |\mathbf{B}|/(t+1)^S \sim |\mathbf{B}|t^{-S}$. □

Figure 7.2 illustrates the convergence rates of both algorithms for the two dimensional parameter case, $\psi \in \mathcal{R}^2$. It shows the iteration histories of both parameter components for deterministic SAEM (solid lines) and EM (dashed lines) for six different pairs of eigenvalues (EV), $(b_1, b_2) \in \{(0.07, 0.03), (0.12, 0.04), (0.21, 0.04), (0.41, 0.04), (0.8, 0.3), (0.94, 0.36)\}$. Notice the different scale in the last four plots.

Figure 7.2 suggests that both algorithms work (almost) equally well, if all eigenvalues of $\mathbf{B}$ are close to zero. However, the performance of deterministic SAEM decreases at a fast pace (relative to that of EM), if at least one of the eigenvalues is significantly larger than zero.

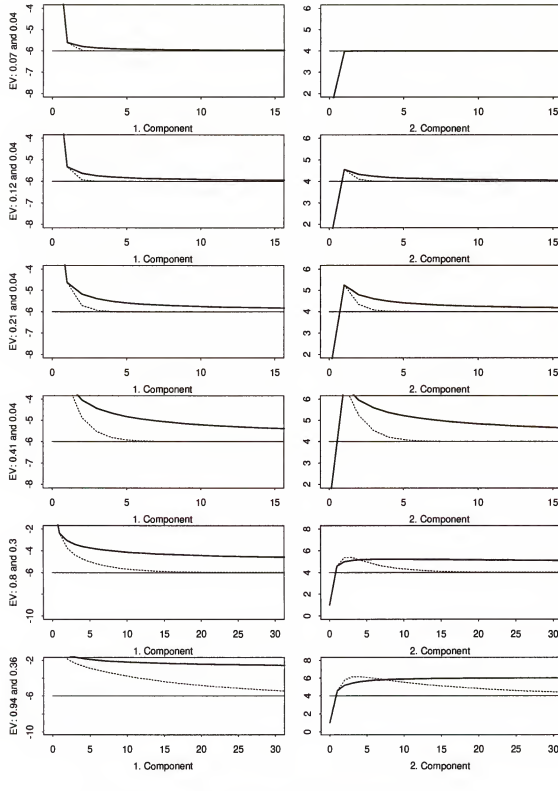

Figure 7.2: Convergence of EM and deterministic SAEM.

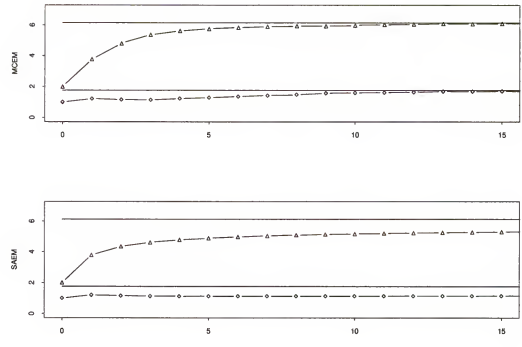

Figure 7.3: MCEM and SAEM for the logistic normal model.

7.2 Examples

We illustrate the convergence behavior of MCEM and SAEM in the following two examples. The first example is the logistic-normal model introduced in Section 5.2.2. The second is an example of a Poisson-gamma model.

7.2.1 Logistic-Normal Model

For the logistic-normal model introduced in Section 5.2.2 (using the "original" data with MLEs $\hat{\beta} = 6.132$ and $\hat{\sigma}^2 = 1.766$) we can use equation (7.12) to calculate the rate matrix. Using numerical integration to evaluate $\mathbf{J}_c$ and, in addition, numerical differentiation to evaluate $\mathbf{J}_m$, we find that the eigenvalues of $\mathbf{B}$ for this model are $\lambda_1 = 0.8143$ and $\lambda_2 = 0.3686$. Since both of the eigenvalues are rather large, we expect SAEM to perform poorly for this model.

Table 7.1: Fish species in lakes

	Number of fish species (y) and area of lake (x)									
y	10	37	60	113	99	13	30	114	112	17
x	5	41	171	25719	59596	1	44	58016	19477	10
y	10	14	39	14	14	67	36	30	19	46
x	85	1	174	3	54	82414	36	1	5	5346
y	68	93	13	53	17	245	88	24	37	22
x	2072	17500	673	2150	2370	28490	4413	29	9065	3302
y	18	214	177	17	50	5	22	156	74	13
x	3626	32893	69484	64500	31500	18500	1125	423488	436000	165
y	11	48	14	28	17	17	21	13	14	21
x	6206	18400	24	10340	2	38000	221	4650	231	7154
y	24	12	26	13	19	19	22	15	9	23
x	616	31153	27195	406	399	1425	60	71	15	98
y	48	21	46	14	7	5	40	18	20	17
x	684	212	676	1080	111	8	8264	9065	357	347

Figure 7.3 shows the first 15 iterations of MCEM and SAEM (using the weights $\gamma_t = 1/(1+t)$) with a constant Monte Carlo sample size, $M = 1000$, per iteration, to estimate β ($\triangle$) and σ^2 ($\diamond$). Using such a large M eliminates random noise in the algorithms almost entirely and thus approximates their deterministic versions well. We observe that MCEM reaches the neighborhood of the MLE after about 13 iterations for both parameter components. However, SAEM at this point is still far from convergence.

7.2.2 Poisson-Gamma Model

Consider the data in Table 7.1, presented in Stein (1988). For 70 lakes of the world the number of fish species (y) and the area of the lake (x) have been recorded. Stein reported that a Poisson log-linear model with $\log(x)$ as the linear predictor results in a deviance of 1538 with 68 degrees of freedom. This indicates that the data is highly over-dispersed relative to Poisson variation. Stein suggests the following Poisson-gamma model to account for the extra variability in the data.

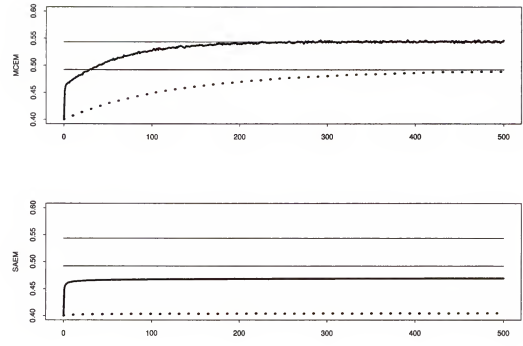

Figure 7.4: MCEM and SAEM for the poisson gamma model.

Suppose that conditional on u_i, $y_i|u_i \sim \text{Poisson}(\mu_i u_i)$, where $\mu_i = \exp\{\beta \log(x_i)\}$. Further suppose that $u_i \sim \text{Gamma}(\kappa, \kappa)$. Then the marginal distribution of y_i is negative binomial with index κ and mean μ_i. Therefore, the likelihood function is available in closed form, resulting in the MLEs $\hat{\beta} = 0.4921$ and $\hat{\kappa} = 0.5436$.

As in the previous example, we calculated the rate matrix and found that the eigenvalues of $\mathbf{B}$ are $\lambda_1 = 0.9936$ and $\lambda_2 = 0.2331$. Since λ_1 is very close to 1, we expect SAEM to perform extremely poorly compared to MCEM.

Figure 7.4 shows the first 500 iterations of MCEM and SAEM (using the weights $\gamma_t = 1/(1+t)$) with a constant Monte Carlo sample size, $M = 1000$, per iteration. The iteration history for β is given by the thick dotted lines and the one for κ by the thick solid lines. Notice that MCEM converges significantly more slowly than in the previous example, reaching the neighborhood of the MLE only after about 250 iterations (for κ) and 500 iterations (for β), respectively. However,

the underlying rate for SAEM is disastrous. In fact, the SAEM iteration history appears to be completely flat in this example.

7.3 Efficiency of MCEM and SAEM

In the previous sections we investigated the convergence rates of MCEM and SAEM. Intuition suggests that an algorithm which converges at a very slow rate makes very inefficient use of the total simulation amount. In the following we investigate the efficiency of MCEM and SAEM empirically.

Consider the balanced, one-way mixed model

$$y_i = \mu + u_i + \epsilon_i\,,\ i = 1, \ldots, m\,, \tag{7.27}$$

where the errors, ϵ_i, are i.i.d. $\mathrm{N}(0,1)$ independently of the random effects, u_i, which are assumed to be i.i.d. $\mathrm{N}(0,\sigma^2)$, with σ^2 known. Model (7.27) is a special case of the OWMM with $r = 1$ and $\sigma_0^2 = 1$. Based on model (7.27) we conducted a simulation study in which we implemented MCEM and SAEM in the following way.

We used the MCEM algorithm with Booth and Hobert's automated way for increasing the Monte Carlo sample size. When the tth step size was swamped by Monte Carlo error we increased the sample size according to $M_{t+1} \leftarrow (1+a)M_t$, using $a = 0.2$. We stopped MCEM and declared convergence when the stopping rule in (4.14) was satisfied for three consecutive iterations, using $\delta_1 = 0.0001$ and $\delta_2 = 0.005$.

Recall that in order to use SAEM, two (rather subjective) decisions have to be made: the choice weight, γ_t, as well as the choice of the Monte Carlo sample size, M, which is held constant over all iterations. In our simulations we implemented SAEM for twelve different combinations of γ_t and M. Specifically, we used four different weights of the form $\gamma_t = A/(A+t)$, $A \in \{1, 10, 100, 500\}$, as well as three different Monte Carlo sample sizes, $M \in \{1, 10, 50\}$. In order to achieve a fair comparison between MCEM and SAEM, we used both algorithms with

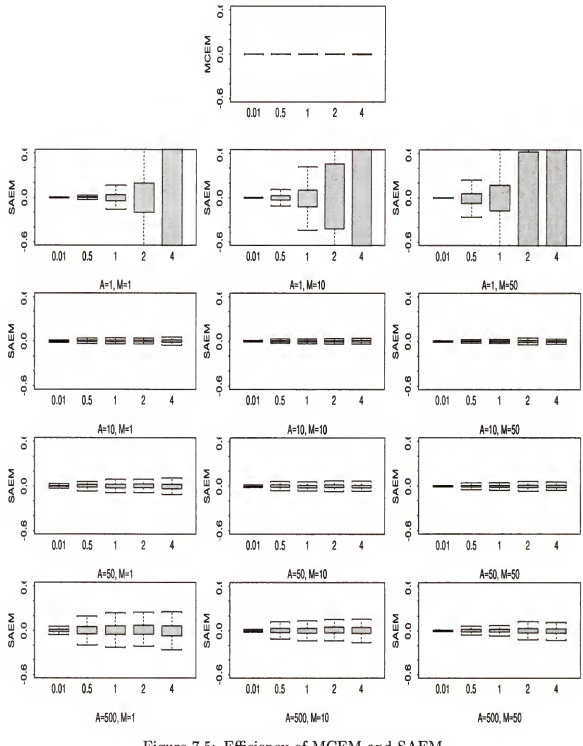

Figure 7.5: Efficiency of MCEM and SAEM

the same total simulation amount; that is, after each run of MCEM we recorded $N = \sum_t M_t$, and then ran each of the twelve variations of SAEM for a total of $\lceil N/M \rceil$ iterations, resulting in the same total simulation amount as for MCEM.

Our simulation study was set up in the following way. For each of the variance components, $\sigma^2 \in \{0.01, 0.5, 1, 2, 4\}$, we simulated 50 sets of data from model (7.27) with the parameters $n = 10$ and $\mu = 0$ held fixed. For each set of data we estimated μ ten times with MCEM and SAEM, using randomly chosen starting values. Figure 7.5 shows box-plots of the *standardized* estimates, that is, the difference between the parameter estimates and the MLE.

Figure 7.5 suggests that MCEM is more efficient than SAEM. Moreover, the performance of SAEM depends strongly on the combination of γ_t and M. For example, for $A = 1$, smaller values of M seem to lead to a better performance. On the other hand, for $A = 500$, larger values of M lead to a more efficient algorithm. Overall, however, both of the values, $A = 1$ and $A = 500$, result in a less efficient algorithm than $A = 10$.

7.4 Conclusion

The appeal of stochastic approximation procedures is their ability to converge with constant Monte Carlo sample size, hence making the decision about whether and when to increase M obsolete.

However, there are clear disadvantages associated with convergence for constant M. We have seen that SAEM typically converges at a much slower rate than MCEM, which can result in a less efficient use of the total simulation amount. Although the performance of SAEM (and also SANR) can be improved by modifying the weight, γ_t, and the (constant) Monte Carlo sample size, M, at the same time, since no guidelines about the optimal choice of γ_t and M exist, implementation of these algorithms is complicated. Furthermore, we have pointed out in Section 4.4.2 that common stopping rules are not appropriate for stochastic

approximation procedures. Thus, the decision to stop the algorithm and declare convergence is often done subjectively, for example, by inspecting a plot of the parameter updates versus the iteration number.

CHAPTER 8
CONCLUSIONS

8.1 Summary of Results

The goal of this dissertation was to compare the performance of stochastic estimation methods for GHMs. We conclude this work with a summary of our most important results.

We started this work by describing the GHM and several of its special cases. We pointed out the benefits of this class of models and indicated its wide range of applications. We argued, however, that the practical use of these models is complicated by the fact that the likelihood function is typically analytically intractable.

We started our discussion of maximum likelihood computation by reviewing several deterministic approaches. We described the ideas of penalized quasi-likelihood and numerical integration and pointed out disadvantages associated with these two approaches. Then we discussed Newton-Raphson and the EM algorithm. We explained the ideas behind these two algorithms and argued that both are typically not directly applicable to GHMs.

This dissertation focused on stochastic approaches for maximum likelihood computation. Among these, we considered five different approaches: SML, which approximates the entire likelihood function, as well as four iterative methods, MCEM, MCNR, SAEM and SANR, which compute only the likelihood mode. Among these four iterative methods we distinguished between methods based on Newton-Raphson and those based on the EM algorithm. We also distinguished between methods that require the Monte Carlo sample size M to be increased

successively for convergence and those which converge with constant M, based on the ideas of stochastic approximation.

Between methods based on EM on those based on Newton-Raphson we found that EM-based methods typically perform better. Indeed, we found that both, MCEM and SAEM, have a much larger domain of attraction than their Newton-Raphson counterparts which greatly simplifies the implementation of these algorithms. In fact, MCEM and SAEM could typically be run with randomly chosen starting values, whereas MCNR and SANR often required a careful search for good starting points. Moreover, we pointed out that both, MCNR and SANR, can become very instable, especially for small Monte Carlo sample sizes. Indeed, we found that for small values of M the Monte Carlo approximation to the Hessian matrix can vary considerably, which frequently causes the algorithms to diverge to the boundary of the parameter space.

Furthermore, in our comparison of MCEM and SAEM we found that MCEM is generally more efficient. Indeed, in Chapter 7 we characterized the convergence rate of both algorithms and found that in GLMMs, MCEM typically converges at a much faster rate. Our simulations support that this fast convergence rate generally results in a much more efficient use of the total simulation amount for MCEM. Moreover, we also found that MCEM is easier to implement than SAEM. Indeed, since for the MCEM algorithm proposed by Booth and Hobert (1999) the decisions when to increase M and when to stop the algorithm are automated, implementation is straightforward. On the other hand, implementation of SAEM is complicated by a set of subjective, user-specific decisions. Prior to starting the algorithm the user has to decide about the choice of the initial Monte Carlo sample size M and the weight γ_t. No guidelines concerning the optimal choice of these two parameters exist. However, we have pointed out that both of these parameters can influence the performance of SAEM dramatically. Furthermore, since there

are no valid stopping rules for stochastic approximation procedures, convergence is often declared rather subjectively, after inspecting a plot of the parameter updates. However, different users will declare convergence at different instances, leading to user-specific differences in the parameter estimates.

Between MCEM and SML, we also found that, generally, MCEM is the more efficient method. Indeed, in Chapter 5 we derived formulae for the asymptotic standard errors of MCEM and SML which show that in most practical applications of GLMMs, SML is very inefficient relative to MCEM. A simulation study and several examples supported these analytical results. In addition, we also pointed out practical difficulties when implementing SML. In particular, since SML requires a numerical routine, like Newton-Raphson, to maximize the simulated likelihood function, good starting values as well as the monitoring of the likelihood are typically necessary to guarantee convergence. Using Quasi-Monte Carlo techniques in an attempt to increase the efficiency of SML, on the other hand, improved its performance only for the cases where the likelihood integrand is sufficiently smooth.

Overall, our investigations suggest that MCEM, for most practical applications, is more efficient and, at the same time, easier to implement than any of the competing methods that we considered in this work. However, some words of cautions are necessary at this point. In this dissertation we restricted our discussion of MCEM to the case of random sampling only, that is, we only considered cases for which it is possible to generate independent and identical samples from the conditional distribution of the random effects given the data. There are, however, many situations for which random sampling is not possible or, at least, not practically feasible. In these situations one would, for example, use a version of MCEM, proposed by McCulloch (1997), which is based on Markov Chain Monte Carlo (MCMC) sampling (that is, dependent sampling). However, Booth and Hobert's arguments for automatically increasing the Monte Carlo sample size are not easily

generalized to the case of dependent samples; in fact, it is not clear at all whether such a generalization can be done (see Booth and Hobert (1999) for a discussion of this issue). One solution to this problem is proposed by Levine and Casella (2001). They suggest to use MCEM with MCMC sampling. In order to apply Booth and Hobert's arguments to their algorithm, they obtain approximate independent and identical samples by sub-sampling the generated MCMC sample during different renewal periods of the Markov Chain. Although their algorithm is certainly a step into the right direction, a lot of work still remains to be done.

8.2 Future Research

A variety of interesting topics for future research still remain. We conclude this dissertation by naming a few of them.

Several questions are still unanswered for MCEM. The parameters α and a in Booth and Hobert's version of MCEM determine the width of the confidence interval and the fraction by which the Monte Carlo sample size should be increased. Booth and Hobert give rather ad hoc recommendations as to how these parameters should be chosen. Investigating the optimal choice for these two parameters is certainly an interesting topic for future work.

A variety of questions are also unanswered for stochastic approximation procedures. Since the choice of the weight γ_t influences the performance of SAEM and SANR drastically, it would certainly be of interest to investigate the optimal choice. We have pointed out that choosing γ_t small already at the early iterations reduces the Monte Carlo error fast, but leads to a slow convergence rate. On the other hand, choosing γ_t large at the start speeds up the convergence but eliminates noise very slowly. A choice of γ_t that balances (in an optimal way) error reduction and the convergence rate would be desirable. Another problem associated with stochastic approximation procedures is the stopping rule. We have seen that common stopping rules, which base the decision to stop on the relative difference

of two successive iterations, often lead to a premature stop of SAEM or SANR. However, there exist, to the best of our knowledge, no alternative stopping rules for stochastic approximation procedures.

This dissertation focused on point estimation only. All of the methods that we considered in this work were compared on the basis of how well they approximate the MLE. Based on this we concluded that MCEM has many advantages over the remaining methods. We did not, however, address the issue of standard error approximation. A method that approximates the parameter well does not necessarily lead to good approximations for its standard error. Gueorguieva (1999), for example, found that the Monte Carlo standard errors of MCEM vary considerably. Moreover, she also found that standard error estimates based on stochastic approximation perform very well. It would certainly be of interest to investigate how the estimation methods considered in this work perform with respect to approximating the standard errors.

A variety of other questions are also yet unsolved. It would also be of interest to compare stochastic estimation methods with non-parametric approaches, which have received a lot of attention in the recent literature. A more thorough investigation of Quasi-Monte Carlo methods and their applicability to MCEM would also be beneficial. And, finally, one of the most important challenges is to develop reliable, fast and user-friendly software for the fitting of hierarchical models. PROC NLMIXED in SAS is one first step into this direction but further extensions are needed. Without the development of widely available software, hierarchical models will only remain a topic for theoretical research without much practical appeal!

APPENDIX A
DERIVATIONS

A.1 Representation for the score function

Recall that the likelihood in (2.6) is defined as $L(\boldsymbol{\psi};\mathbf{y}) = \int f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})d\mathbf{u} = f(\mathbf{y};\boldsymbol{\psi})$. Thus, under regularity conditions that allow for interchanging integration and differentiation, we get for the score function

$$\begin{aligned}
\mathbf{S}(\boldsymbol{\psi}) &= \frac{\partial}{\partial\boldsymbol{\psi}} \log \int f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})d\mathbf{u} \\
&= \frac{\partial}{\partial\boldsymbol{\psi}} \int f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})d\mathbf{u} \Big/ \int f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})d\mathbf{u} \\
&= \int \left[\frac{\partial}{\partial\boldsymbol{\psi}} f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})\right] \frac{f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})}{f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})} d\mathbf{u} \Big/ f(\mathbf{y};\boldsymbol{\psi}) \\
&= \int \left[\frac{\partial}{\partial\boldsymbol{\psi}} \log f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})\right] \frac{f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})}{f(\mathbf{y};\boldsymbol{\psi})} d\mathbf{u} \\
&= E\left[\frac{\partial}{\partial\boldsymbol{\psi}} \log f(\mathbf{y},\mathbf{u};\boldsymbol{\psi}) \middle| \mathbf{y};\boldsymbol{\psi}\right].
\end{aligned}$$

But this implies that $\mathbf{S}(\boldsymbol{\psi}) = \mathbf{F}(\boldsymbol{\psi},\boldsymbol{\psi})$, with $\mathbf{F}$ defined in (3.15).

A.2 Representation for the Hessian matrix

Recall that the Hessian matrix in (3.7) is defined as $\mathbf{H}(\boldsymbol{\psi}) = \partial^2 \log L(\boldsymbol{\psi};\mathbf{y})/\partial\boldsymbol{\psi}\partial\boldsymbol{\psi}' = \partial\mathbf{S}(\boldsymbol{\psi};\mathbf{y})/\partial\boldsymbol{\psi}'$. Using Section A.1, we get

$$\begin{aligned}
\mathbf{H}(\boldsymbol{\psi}) &= \frac{\partial}{\partial\boldsymbol{\psi}'} \left(\int \left[\frac{\partial}{\partial\boldsymbol{\psi}} \log f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})\right] \frac{f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})}{f(\mathbf{y};\boldsymbol{\psi})} d\mathbf{u} \right) \\
&= I_1 + \underbrace{\int \left[\frac{\partial}{\partial\boldsymbol{\psi}} \log f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})\right] \frac{\partial}{\partial\boldsymbol{\psi}'} \left(\frac{f(\mathbf{y},\mathbf{u};\boldsymbol{\psi})}{f(\mathbf{y};\boldsymbol{\psi})} \right) d\mathbf{u}}_{\equiv J}, \qquad \text{(A.1)}
\end{aligned}$$

where

$$I_1 \equiv E\left(\frac{\partial^2}{\partial\psi\partial\psi'}\log f(\mathbf{y},\mathbf{u};\psi)\middle|\mathbf{y};\psi\right) = \mathbf{H}_Q(\psi),$$

where $\mathbf{H}_Q(\psi)$ is the Hessian of the Q-function in equation (3.23). Straightforward calculus shows that

$$\frac{\partial}{\partial\psi'}\left(\frac{f(\mathbf{y},\mathbf{u};\psi)}{f(\mathbf{y};\psi)}\right) = g(\mathbf{u}|\mathbf{y};\psi)\left(\frac{\partial}{\partial\psi'}\log f(\mathbf{y},\mathbf{u};\psi) - \frac{\partial}{\partial\psi'}\log f(\mathbf{y};\psi)\right),$$

so we can write for the term on the right hand side of equation (A.1)

$$J = I_2 - I_3, \tag{A.2}$$

where

$$\begin{aligned} I_2 &\equiv E\left(\left[\frac{\partial}{\partial\psi}\log f(\mathbf{y},\mathbf{u};\psi)\right]\left[\frac{\partial}{\partial\psi}\log f(\mathbf{y},\mathbf{u};\psi)\right]'\middle|\mathbf{y};\psi\right), \\ I_3 &\equiv E\left(\frac{\partial}{\partial\psi}\log f(\mathbf{y},\mathbf{u};\psi)\middle|\mathbf{y};\psi\right)E\left(\frac{\partial}{\partial\psi}\log f(\mathbf{y},\mathbf{u};\psi)\middle|\mathbf{y};\psi\right)'. \end{aligned}$$

Thus the Hessian can be written as

$$\mathbf{H}(\psi) = I_1 + I_2 - I_3.$$

Now, since

$$\mathrm{Var}\left(\frac{\partial}{\partial\psi}\log f(\mathbf{y},\mathbf{u};\psi)\middle|\mathbf{y};\psi\right) = I_2 - I_3,$$

an equivalent representation for the Hessian is

$$\mathbf{H}(\psi) = \mathbf{H}_Q(\psi) + \mathrm{Var}\left(\frac{\partial}{\partial\psi}\log f(\mathbf{y},\mathbf{u};\psi)\middle|\mathbf{y};\psi\right).$$

A.3 Derivations for the LMM

Recall that for the LMM defined in Section 2.1.2,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}, \tag{A.3}$$

with $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{G})$, independent of $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}^{-1})$. Standard manipulations of multivariate normal distributions show that

$$\mathbf{u}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{A.4}$$

where $\boldsymbol{\Sigma} = [\mathbf{Z}'\mathbf{W}\mathbf{Z} + \mathbf{G}^{-1}]^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}\mathbf{Z}'\mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$.

Specifically, in the OWMM we have, $\mathbf{G} = \sigma_1^2\mathbf{I}_m$, $\mathbf{W} = (1/\sigma_0^2)\mathbf{I}_n$, and $\mathbf{Z}_{n\times m} = \mathbf{I}_m \otimes \mathbf{1}_r$, where $n = m \times r$. Thus, equation (A.4) implies that for the OWMM,

$$\mathbf{u}|\mathbf{y} \sim \mathcal{N}\left(\frac{1}{1+\rho}(\bar{y}_{\cdot} - \mu), \frac{\sigma_0^2}{r}\frac{1}{1+\rho}\mathbf{I}_m\right), \tag{A.5}$$

where $\rho = \sigma_0^2/(r\sigma_1^2)$ and $\bar{\mathbf{y}}_{\cdot} = (\bar{y}_{1\cdot}, \ldots, \bar{y}_{m\cdot})'$.

In the following we derive the EM parameter update for the LMM, assuming that $\mathbf{W}$ and $\mathbf{G}$ are known and that we are only interested in estimating $\boldsymbol{\beta}$. As in Section 5.1 we write $f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta})$ for the conditional density of the data. It follows that

$$\frac{\partial}{\partial\boldsymbol{\beta}} \log f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta}) = \mathbf{X}'\mathbf{W}[\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{u}]. \tag{A.6}$$

Using (A.4), this implies that the EM estimating equations for $\boldsymbol{\beta}$ solve

$$\mathbf{0} = \mathbf{F}(\boldsymbol{\beta};\boldsymbol{\beta}^{(t)}) = \mathbf{X}'\mathbf{W}[\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\boldsymbol{\mu}^{(t)}], \tag{A.7}$$

where $\boldsymbol{\mu}^{(t)} = \boldsymbol{\Sigma}\mathbf{Z}'\mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{(t)})$. Thus, the $(t+1)$st EM update satisfies

$$\mathbf{X}'\mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{(t+1)}) = \mathbf{X}'\mathbf{W}\mathbf{A}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{(t)}), \tag{A.8}$$

where $\mathbf{A} = \mathbf{Z}\boldsymbol{\Sigma}\mathbf{Z}'\mathbf{W}$. On the other hand, one can show that the MLE, $\hat{\boldsymbol{\beta}}$, satisfies

$$\mathbf{X}'\mathbf{W}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{X}'\mathbf{W}\mathbf{A}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}). \tag{A.9}$$

Subtracting equation (A.8) from (A.9) gives

$$\boldsymbol{\beta}^{(t+1)} - \hat{\boldsymbol{\beta}} = \mathbf{B}(\boldsymbol{\beta}^{(t)} - \hat{\boldsymbol{\beta}}), \tag{A.10}$$

where the rate matrix in the LMM is

$$\mathbf{B} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}(\mathbf{X}'\mathbf{W}\mathbf{Z}\boldsymbol{\Sigma}\mathbf{Z}'\mathbf{W}\mathbf{X}). \tag{A.11}$$

A.4 Estimating μ in the OWMM

Consider the OWMM,

$$y_{ij} = \mu + u_i + \epsilon_{ij}, \quad (i = 1, \ldots, m; j = 1, \ldots, r), \tag{A.12}$$

introduced in Section 2.1.3. Recall that ϵ_{ij} and u_i are a random sample from $N(0, \sigma_0^2)$ and $N(0, \sigma_1^2)$, respectively. We assume that the variance components σ_0^2 and σ_1^2 are known and that we are only interested in estimating the mean μ. The complete data likelihood is given by

$$f(\mathbf{y}, \mathbf{u}; \mu) = \frac{\exp\left[-\sum_{i=1}^{m}\sum_{j=1}^{r}(y_{ij} - u_i - \mu)^2/(2\sigma_0^2)\right]}{(2\pi\sigma_0^2)^{mr/2}} \frac{\exp\left[-\sum_{i=1}^{m} u_i^2/(2\sigma_1^2)\right]}{(2\pi\sigma_1^2)^{m/2}}. \tag{A.13}$$

It follows from (A.13) that

$$\frac{\partial}{\partial\mu}\log f(\mathbf{y}, \mathbf{u}; \mu) = \frac{mr}{\sigma_0^2}(\hat{\mu} - \bar{u}_\cdot - \mu), \tag{A.14}$$

where $\bar{u}_\cdot$ denotes the sample mean of the u_i's and $\hat{\mu} = \bar{y}_{\cdot\cdot}$ denotes the MLE of μ. Using equation (A.5) from Section A.3, we get for the conditional distribution of $\bar{u}_\cdot$

$$\bar{u}_\cdot|\mathbf{y}; \mu' \sim N\left(b(\hat{\mu} - \mu'), \frac{b\sigma_0^2}{mr}\right), \tag{A.15}$$

where $b = 1/(1+\rho) = (r\sigma_1^2)/(r\sigma_1^2 + \sigma_0^2)$. It follows from (A.14) and (A.15) that

$$E\left[\frac{\partial}{\partial\mu}\log f(\mathbf{y}, \mathbf{u}; \mu)\middle|\mathbf{y}; \mu'\right] = \frac{mr}{\sigma_0^2}\left[(1-b)\hat{\mu} + b\mu' - \mu\right]. \tag{A.16}$$

Recall that the EM update satisfies $F(\mu, \mu') = 0$, with F defined in equation (3.15). Using (A.16), we get

$$F(\mu, \mu') \propto (1-b)\hat{\mu} + b\mu' - \mu; \tag{A.17}$$

thus, the EM parameter update is given by $\mu = (1-b)\hat{\mu} + b\mu'$. Using Section A.1 and equation (A.16), it follows that for the score function

$$S(\mu) = -\frac{mr}{\sigma_0^2}(1-b)(\mu - \hat{\mu}). \tag{A.18}$$

Differentiating (A.18), we get for the Hessian

$$H(\mu) = -\frac{mr}{\sigma_0^2}(1-b). \tag{A.19}$$

Now consider the Monte Carlo approximations to the Q-function and the score function. Using (A.14) we get

$$\frac{1}{M}\sum_{k=1}^{M}\frac{\partial}{\partial\mu}\log f(\mathbf{y}, \mathbf{u}^{(k)}; \mu) = \frac{mr}{\sigma_0^2}(\hat{\mu} - \bar{\bar{u}}_\cdot - \mu), \tag{A.20}$$

where $\mathbf{u}^{(1)}, ..., \mathbf{u}^{(M)} \overset{iid}{\sim} \mathbf{u}|\mathbf{y}; \mu'$ and $\bar{\bar{u}}_\cdot = \sum_{k=1}^{M} \bar{u}^k/M$. It follows from equation (A.5) in Section A.3 that

$$\bar{\bar{u}}_\cdot|\mathbf{y}; \mu' \sim b(\hat{\mu} - \mu') + e, \tag{A.21}$$

where $e \sim \mathcal{N}(0, b\sigma_0^2/(Mmr))$. Recall that the MCEM update solves $\tilde{F}(\mu, \mu') = 0$, with $\tilde{F}$ defined in (4.10). It follows from (A.20) and (A.21) that

$$\tilde{F}(\mu, \mu') \propto -\mu + (1-b)\hat{\mu} + b\mu' + e; \tag{A.22}$$

thus, the MCEM parameter update is $\mu = (1-b)\hat{\mu} + b\mu' + e$. Using similar arguments, the Monte Carlo score function in (4.15) is

$$\tilde{S}(\mu) = \frac{mr}{\sigma_0^2}[(1-b)(\hat{\mu} - \mu) + e]. \tag{A.23}$$

A.5 Approximating $\boldsymbol{\tau}^2_{\text{MCEM}}$ and $\boldsymbol{\tau}^2_{\text{SML}}$

Recall the situation from Section 5.1.1. Since the unknown parameter $\boldsymbol{\beta}$ enters the density of the complete data, $f(\mathbf{y}, \mathbf{u}; \boldsymbol{\beta})$, only through $f(\mathbf{y}|\mathbf{u}; \boldsymbol{\beta})$, the gradient and Hessian of $\log f(\mathbf{y}|\mathbf{u}; \boldsymbol{\beta})$, emphasizing the dependence on the random effects $\mathbf{u}$,

are

$$\mathbf{h}_1(\mathbf{u}) \equiv \frac{\partial}{\partial \boldsymbol{\beta}} \log f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{y_i - \mu_i}{a_i V(\mu_i) g'(\mu_i)} \mathbf{x}_i \tag{A.24}$$

$$\mathbf{h}_2(\mathbf{u}) \equiv \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} \log f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta}) = -\sum_{i=1}^{n} \frac{1}{a_i V(\mu_i) g'(\mu_i)^2} \mathbf{x}_i \mathbf{x}_i' - \mathbf{R}_\beta. \tag{A.25}$$

Notice that $\mathbf{R}_\beta = 0$ if g is the canonical link function, because in that case $V(\mu)g'(\mu) \equiv 1$. In non-canonical models, $\mathbf{R}_\beta = O_p(n^{1/2})$ and may not be negligible relative to the other terms in (A.25). Let $\mathbf{W}$ be the diagonal matrix of iterative GLM weights, $w_{ii} = 1/\{a_i V(\mu_i) g'(\mu_i)^2\}$. We approximate (A.24) and (A.25) by

$$\mathbf{h}_1(\mathbf{u}) \approx \mathbf{X}'\mathbf{W}(g(\mathbf{y}) - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{u}), \tag{A.26}$$

$$\mathbf{h}_2(\mathbf{u}) \approx -\mathbf{X}'\mathbf{W}\mathbf{X}, \tag{A.27}$$

where we used a first order approximation of $g(y_i)$ about μ_i in (A.26). We note that in the linear mixed model, $\mathbf{W} = (1/\sigma_0^2)\mathbf{I}$ and (A.26) and (A.27) hold exactly.

Let $\hat{c}$ be the normalizing factor of the posterior distribution of $\mathbf{u}$ evaluated at the MLE,

$$\hat{c} = \int f(\mathbf{y}, \mathbf{u}; \hat{\boldsymbol{\beta}}) d\mathbf{u}. \tag{A.28}$$

With the notation defined in (A.24) and (A.25), we can write the integrals in (5.1), (5.3), (5.5) and (5.6) as

$$\hat{\mathbf{I}} = E\left[\mathbf{h}_1(\mathbf{u})\mathbf{h}_1(\mathbf{u})'|\mathbf{y};\boldsymbol{\beta}\right]|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}, \tag{A.29}$$

$$\hat{\mathbf{J}} = E[\mathbf{h}_2(\mathbf{u})|\mathbf{y};\boldsymbol{\beta}]|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}, \tag{A.30}$$

$$\tilde{\mathbf{I}} = \int \mathbf{h}_1(\mathbf{u})\mathbf{h}_1(\mathbf{u})' \{f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta})\}^2 f(\mathbf{u}) d\mathbf{u}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}, \tag{A.31}$$

$$\tilde{\mathbf{J}} = \int \left[\mathbf{h}_1(\mathbf{u})\mathbf{h}_1(\mathbf{u})' + \mathbf{h}_2(\mathbf{u})\right] f(\mathbf{y}|\mathbf{u};\boldsymbol{\beta}) f(\mathbf{u}) d\mathbf{u}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} = \hat{c}\left\{\hat{\mathbf{I}} + \hat{\mathbf{J}}\right\}. \tag{A.32}$$

Using the Laplace approximation (De Bruijn, 1958), leads to simple approximate formulæ for (A.29), (A.30), (A.31) and (A.32). Specifically, we use arguments very

similar to those outlined in Booth and Hobert (1998, Sec.4). To simplify notation, we suppress the dependence on $\mathbf{y}$ and $\boldsymbol{\beta}$ and write the joint density of the complete data, evaluated at the MLE, as $f(\mathbf{y},\mathbf{u};\hat{\boldsymbol{\beta}}) = \exp\{\hat{l}(\mathbf{u})\}$ and let $\hat{l}^{(r)}(\mathbf{u}) = \partial^r \hat{l}(\mathbf{u})/\partial \mathbf{u}^r$, for $r = 1, 2$. Let $\hat{\mathbf{u}}$ denote the maximizer of $\hat{l}(\mathbf{u})$, satisfying $\hat{l}^{(1)}(\mathbf{u}) = 0$. Let $\hat{\mathbf{W}}$ be the matrix of iterative weights evaluated at $\mathbf{u} = \hat{\mathbf{u}}$. Then the Laplace approximation to $\hat{c}$ in (A.28) becomes

$$\hat{c} = \int \exp\{\hat{l}(\mathbf{u})\} d\mathbf{u} \approx |-2\pi \hat{l}^{(2)}(\hat{\mathbf{u}})^{-1}|^{1/2} \exp\{\hat{l}(\hat{\mathbf{u}})\}. \tag{A.33}$$

Booth and Hobert (1998, Eq.12&14) show that

$$\hat{\mathbf{u}} \approx E[\mathbf{u}|\mathbf{y};\hat{\boldsymbol{\beta}}] \text{ and } -\hat{l}^{(2)}(\hat{\mathbf{u}})^{-1} \approx \mathrm{Var}[\mathbf{u}|\mathbf{y};\hat{\boldsymbol{\beta}}] \approx \boldsymbol{\Sigma}, \tag{A.34}$$

with $\boldsymbol{\Sigma}$ defined in (5.7). In general, these approximations have a relative error of $O_p(n^{-1})$ and will perform well if n is large (Tierney and Kadane, 1986). Notice that $\exp\{\hat{l}(\hat{\mathbf{u}})\} = f(\mathbf{y}|\hat{\mathbf{u}};\hat{\boldsymbol{\beta}}) f(\hat{\mathbf{u}})$, where $f(\hat{\mathbf{u}}) = |2\pi\mathbf{G}|^{-1/2} \exp\{-\hat{\mathbf{u}}'\mathbf{G}^{-1}\hat{\mathbf{u}}/2\}$. Using the approximation for $-(\hat{l}^{(2)})^{-1}$ in (A.34) and omitting constants from $\exp\{\hat{l}(\hat{\mathbf{u}})\}$ that involve the data only, (A.33) becomes

$$\hat{c} \approx |\boldsymbol{\Sigma}|^{1/2} |\mathbf{G}|^{-1/2} \exp\{-\hat{\mathbf{u}}'\mathbf{G}^{-1}\hat{\mathbf{u}}/2\}. \tag{A.35}$$

Observe now that for the expectation in (A.29) we can write

$$\hat{\mathbf{I}} = \mathrm{Var}\,[\mathbf{h}_1(\mathbf{u})|\mathbf{y};\boldsymbol{\beta}]|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} + \boldsymbol{\mu}_{h_1}\boldsymbol{\mu}_{h_1}', \tag{A.36}$$

where $\boldsymbol{\mu}_{h_1} \equiv E\,[\mathbf{h}_1(\mathbf{u})|\mathbf{y};\boldsymbol{\beta}]|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}$. The variance term on the right hand side of (A.36) is of order n. In contrast the product term is identically zero in the LMM and generally of smaller order than the variance term. Using (A.26), we get $\mathrm{Var}\,[\mathbf{h}_1(\mathbf{u})|\mathbf{y};\boldsymbol{\beta}]\,|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} \approx \mathbf{X}'\hat{\mathbf{W}}\mathbf{Z}\ \mathrm{Var}[\mathbf{u}|\mathbf{y};\boldsymbol{\beta}]\ \mathbf{Z}'\hat{\mathbf{W}}\mathbf{X}$ and with (A.34) we approximate (A.36) by

$$\hat{\mathbf{I}} \approx \mathbf{X}'\hat{\mathbf{W}}\mathbf{Z}\boldsymbol{\Sigma}\mathbf{Z}'\hat{\mathbf{W}}\mathbf{X}. \tag{A.37}$$

Similarly, using (A.27) and (A.30), we obtain

$$\hat{\mathbf{J}} \approx -\mathbf{X}'\hat{\mathbf{W}}\mathbf{X}, \tag{A.38}$$

which, combined with (A.37), gives

$$\boldsymbol{\tau}^2_{\text{MCEM}} \approx (\mathbf{X}'\hat{\mathbf{W}}\mathbf{X})^{-1}\mathbf{X}'\hat{\mathbf{W}}\mathbf{Z}\ \boldsymbol{\Sigma}\ \mathbf{Z}'\hat{\mathbf{W}}\mathbf{X}(\mathbf{X}'\hat{\mathbf{W}}\mathbf{X})^{-1}, \tag{A.39}$$

which is exact in the LMM. Omitting constant terms, the *generalized variance* of the MCEM error becomes

$$|\boldsymbol{\tau}^2_{\text{MCEM}}| \propto |\boldsymbol{\Sigma}|, \tag{A.40}$$

where $|\cdot|$ denotes the determinant of a quadratic matrix.

We will use similar arguments to approximate the asymptotic variance $\boldsymbol{\tau}^2_{\text{SML}}$. This approximation will again hold exactly in the LMM. Using (A.35), (A.37) and (A.38), it follows that (A.32) can be approximated by

$$\tilde{\mathbf{J}} \approx |\mathbf{G}|^{-1/2}|\boldsymbol{\Sigma}|^{1/2}\exp\{-\tilde{\mathbf{u}}'\mathbf{G}^{-1}\tilde{\mathbf{u}}/2\}\ (\mathbf{X}'\hat{\mathbf{W}}\mathbf{Z}\boldsymbol{\Sigma}\mathbf{Z}'\hat{\mathbf{W}}\mathbf{X} - \mathbf{X}'\hat{\mathbf{W}}\mathbf{X}). \tag{A.41}$$

In order to approximate $\tilde{\mathbf{I}}$ in (A.31), we will use similar arguments but we need a slightly different notation from before. Let us write $\{f(\mathbf{y}|\mathbf{u};\hat{\boldsymbol{\beta}})\}^2\ f(\mathbf{u})$ $= \exp\{\tilde{l}(\mathbf{u})\}$ and define $\tilde{l}^{(\mathbf{r})}(\mathbf{u})$, $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{W}}$ in the same way as before. We define the normalizing constant $\tilde{c} = \int \exp\{\tilde{l}(\mathbf{u})\}d\mathbf{u}$ and the density $\tilde{f}(\mathbf{u}) = \exp\{\tilde{l}(\mathbf{u})\}/\tilde{c}$. Similar to (A.34) and (A.35), the Laplace approximation to the normalizing constant is $\tilde{c} \approx |\tilde{\boldsymbol{\Sigma}}|^{1/2}|\mathbf{G}|^{-1/2}\ \exp\{-\tilde{\mathbf{u}}'\mathbf{G}^{-1}\tilde{\mathbf{u}}/2\}$ and the approximations to the mean and variance (with respect to the density $\tilde{f}$) are

$$\tilde{\mathbf{u}} \approx E_{\tilde{f}}[\mathbf{u}] \text{ and } -\tilde{l}^{(2)}(\tilde{\mathbf{u}})^{-1} \approx \text{Var}_{\tilde{f}}[\mathbf{u}] \approx \tilde{\boldsymbol{\Sigma}}, \tag{A.42}$$

with $\tilde{\boldsymbol{\Sigma}} = [2\ \mathbf{Z}'\tilde{\mathbf{W}}\mathbf{Z} + \mathbf{G}^{-1}]^{-1}$ in contrast to (5.7). It follows that the integral in (A.31) becomes

$$\tilde{\mathbf{I}} = \tilde{c}\left\{\text{Var}_{\tilde{f}}[\mathbf{h}_1(\mathbf{u})]\big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} + \tilde{\boldsymbol{\mu}}_{h_1}\tilde{\boldsymbol{\mu}}'_{h_1}\right\}, \tag{A.43}$$

where $\tilde{\boldsymbol{\mu}}_{h_1} = E_{\tilde{f}}[\mathbf{h}_1(\mathbf{u})]|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}}$. In similar fashion we find the approximation of (A.43) to be

$$\tilde{\mathbf{I}} \approx |\mathbf{G}|^{-1/2}|\tilde{\boldsymbol{\Sigma}}|^{1/2}\exp\{-\tilde{\mathbf{u}}'\mathbf{G}^{-1}\tilde{\mathbf{u}}/2\}\left\{\mathbf{X}'\tilde{\mathbf{W}}\mathbf{Z}\tilde{\boldsymbol{\Sigma}}\mathbf{Z}'\tilde{\mathbf{W}}\mathbf{X}\right\}. \tag{A.44}$$

We have argued before, that $|\boldsymbol{\Sigma}|$ (and therefore also $|\tilde{\boldsymbol{\Sigma}}|$) are bounded. Furthermore, for fixed values of $\mathbf{u}$, $\exp\{-\mathbf{u}'\mathbf{G}^{-1}\mathbf{u}/2\}$ is bounded between 0 and 1. Combining (A.41) and (A.44), it follows that, for the generalized variance of the Monte Carlo error of SML,

$$|\boldsymbol{\tau}^2_{\text{SML}}| = O_p\left(|\mathbf{G}|^{1/2}\right), \tag{A.45}$$

which completes the proof.

APPENDIX B
OX PROGRAM CODE

The following is the essential OX program code to fit model (7.27) with automated MCEM and SAEM, using the same total simulation amount.

```
main(){//Begin of Program
decl ... //Define all variables
eps=0.0001; delta=0.003;//convergence constants
n=10;B=0.5;mle=1;//model parameters
startM=10;//initial Monte Carlo sample size
a=0.3;//fraction by which sample size increased
A=1;//weight for SAEM
start=0;//starting value for algorithm
//***Start MCEM
mcem=start;N=0;conseq=0;converge=0;M=startM;N=0;//initialize
while (converge !=1)
{ cmean=B*(mle-mcem);cvar=B/n;//conditional mean and variance
  u=(rann(1,M).*sqrt(cvar))+cmean;//draw MC sample
  oldmcem=mcem;mcem=mle-sumr(u)/M;//update parameter
  N=N+M;//record total simulation amount
  //now we update MC sample size
  varhat=sumr((mle-u-mcem).^2)/M^2;chisq=(oldmcem-mcem)^2/varhat;
  if (chisq< quanchi(0.75,1)&&conseq ==0){M=M+trunc(M*a);}
  //now we diagnose convergence
  if(norm((mcem-oldmcem)/(oldmcem+eps))< delta)
```

```
  {if(conseq!=0){conseq=conseq+1;if(conseq==3){converge=1;}}
   else {conseq=1;}}else{conseq=0;}
}
//***Start SAEM
r=1;gamma=A/(A+r);saem=start;curu=0;cumu=0;M=startM;//initialize
while (r<=trunc(N/M))
{ cmean=B*(mle-saem);cvar=B/n;//conditional mean and variance
  u=(rann(1,M).*sqrt(cvar))+cmean;//draw MC sample
  curu=gamma*(sumr(u)/M);//weighted MC sample from current iteration
  cumu=(1-gamma)*cumu;//weighted MC sample from past iterations
  oldsaem=saem;saem=mle-(cumu+curu);//update parameter
  cumu=cumu+curu;//update past MC sample
  gamma=A/(A+r);//increase weight
  r=r+1;//increase iteration number
}
}//End of Program
```

REFERENCES

Abramowitz, M. and Stegun, I. A., editors (1992), *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Dover Publications.

Agresti, A., Booth, J. G., Hobert, J. P., and Caffo, B. (2001), "Random Effects Modeling of Categorical Response Data," *Sociological Methodology*, In press.

Aitkin, M. (1996), "A General Maximum Likelihood Analysis of Overdispersion in Generalized Linear Models," *Statistics and Computing*, 6, 251–262.

Aitkin, M. (1999), "A General Maximum Likelihood Analysis of Variance Components in Generalized Linear Mixed Models," *Biometrics*, 55, 117–128.

Anderson, D. A. and Aitkin, M. (1985), "Variance Component Models With Binary Response: Interviewer Variability," *Journal of the Royal Statistical Society B*, 47, 203–210.

Bard, Y. (1974), *Nonlinear Parameter Estimation*, New York: Academic Press.

Blum, J. R. (1954), "Approximation Methods which converge with Probability One," *The Annals of Mathematical Statistics*, 25, 382–386.

Booth, J. G. and Caffo, B. (2001), "Unequal Sampling for Monte Carlo EM Algorithms," Technical report, University of Florida, Department of Statistics.

Booth, J. G. and Hobert, J. P. (1998), "Standard Errors of Prediction in Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 93, 262–272.

Booth, J. G. and Hobert, J. P. (1999), "Maximizing Generalized Linear Mixed Model Likelihoods with an Automated Monte Carlo EM Algorithm," *Journal of the Royal Statistical Society B*, 61, 265–285.

Bouleau, N. and Lépingle, D. (1994), *Numerical Methods for Stochastic Processes*, New York: Wiley.

Breslow, N. E. and Clayton, D. G. (1993), "Approximate Inference in Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 88, 9–25.

Breslow, N. E. and Lin, X. (1995), "Bias Correction in Generalised Linear Mixed Models With a Single Component of Dispersion," *Biometrika*, 82, 81–91.

Bronstein, I. N. and Semendjajew, K. A. (1991), *Taschenbuch der Mathematik*, Stuttgart: Teubner.

Callanan, T. P. and Harville, D. A. (1991), "Some New Algorithms for Computing Restricted Maximum Likelihood Estimates of Variance Components," *Journal of Statistical Computation and Simulation*, 38, 239–259.

Carletti, M., Pallini, A., and Pesarin, F. (1994), "Quasi-Monte Carlo Integration Methods for the Bootstrap," *Proceedings of the Italian Statistical Society*, 2, 441–448.

Chan, K. S. and Ledolter, J. (1995), "Monte Carlo EM Estimation for Time Series Models Involving Counts," *Journal of the American Statistical Association*, 90, 242–252.

Conaway, M. R. (1990), "A Random Effects Model for Binary Data," *Biometrics*, 46, 317–328.

Crepon, B. and Duguet, E. (1997), "Research and Development, Competition and Innovation: Pseudo-maximum Likelihood and Simulated Maximum Likelihood Methods Applied to Count Data Models With Heterogeneity," *Journal of Econometrics*, 79, 355–378.

Crouch, E. A. C. and Spiegelman, D. (1990), "The Evaluation of Integrals of the Form $\int_{-\infty}^{+\infty} f(t)\exp(-t^2)\,dt$: Application to Logistic-normal Models," *Journal of the American Statistical Association*, 85, 464–469.

Danielsson, J. (1994), "Stochastic Volatility in Asset Prices. Estimation With Simulated Maximum Likelihood," *Journal of Econometrics*, 64, 375–400.

De Bruijn, N. G. (1958), *Asymptotic Methods in Analysis*, Amsterdam: North-Holland.

Delyon, B., Lavielle, M., and Moulines, E. (1999), "Convergence of a Stochastic Approximation version of the EM Algorithm," *The Annals of Statistics*, 27, 94–128.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood From Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society B*, 39, 1–22.

Evans, M. and Swartz, T. (1995), "Methods for Approximating Integrals in Statistics With Special Emphasis on Bayesian Integration Problems," *Statistical Science*, 10, 254–272.

Fang, K.-T. and Wang, Y. (1994), *Number Theoretic Methods in Statistics*, New York: Chapman & Hall.

Fang, K.-T., Wang, Y., and Bentler, P. M. (1994), "Some Applications of Number-theoretic Methods in Statistics," *Statistical Science*, 9, 416–428.

Faure, H. (1982), "Discrépance de Suites associées à un Système de Numération (en dimension s)," *Acta Arithmetica*, 41, 337–351.

Follmann, D. A. and Lambert, D. (1989), "Generalizing Logistic Regression By Nonparametric Mixing," *Journal of the American Statistical Association*, 84, 295–300.

Friedl, H. (1998), "Nichtparametrische Maximum Likelihood Schaetzung in Generalisierten Linearen Mischmodellen," *Oesterreichische Zeitschrift fuer Statistik*, 26, 7–30.

Gauderman, W. and Navidi, W. (2001), "A Monte Carlo Newton-Raphson Procedure for Maximizing Complex Likelihoods on Pedigree Data," *Computational Statistics and Data Analysis*, 35, 395–415.

Gelfand, A. E. and Carlin, B. P. (1993), "Maximum-likelihood Estimation for Constrained- Or Missing-data Models," *Canadian Journal of Statistics*, 21, 303–311.

Geyer, C. J. and Thompson, E. A. (1992), "Constrained Monte Carlo Maximum Likelihood for Dependent Data," *Journal of the Royal Statistical Society B*, 54, 657–683.

Gilmour, A. R., Anderson, R. D., and Rae, A. L. (1985), "The Analysis of Binomial Data By a Generalized Linear Mixed Model," *Biometrika*, 72, 593–599.

Gouriéroux, C. and Monfort, A. (1993), "Simulation-Based Inference: A Survey with special Reference to Panel Data Models," *Journal of Econometrics*, 59(1-2), 5–33.

Gu, M. G. and Li, S. (1998), "A Stochastic Approximation Algorithm for Maximum Likelihood Estimation with Incomplete Data," *Canadian Journal of Statistics*, 26, 567–582.

Gueorguieva, R. (1999), *Models for Repeated Measures of a Multivariate Response*, PhD dissertation, University of Florida, Department of Statistics.

Györfi, L. and Walk, H. (1996), "On the Averaged Stochastic Approximation for Linear Regression," *SIAM Journal on Control and Optimization*, 34, 31–61.

Halton, J. H. (1960), "On the Efficiency of certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals," *Numerische Mathematik*, 2, 84–90.

Heckman, J. and Singer, B. (1984), "A Method for Minimizing the Impact of Distributional Assumptions in Econometric Models for Duration Data," *Econometrica*, 52, 271–320.

Hinde, J. and Demetrio, C. (1998), "Overdispersion: Models and Estimation," *Computational Statistics and Data Analysis*, 27, 151–170.

Hobert, J. P. (2000), "Hierarchical Models: A Current Computational Perspective," *Journal of the American Statistical Association*, 95, 1312–1316.

Jamshidian, M. and Jennrich, R. I. (1993), "Conjugate Gradient Acceleration of the EM Algorithm," *Journal of the American Statistical Association*, 88, 221–228.

Jiang, J. (1998), "Consistent Estimators in Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 93, 720–729.

Kallianpur, G. (1954), "A Note on the Robbins-Monro Stochastic Approximation Method," *The Annals of Mathematical Statistics*, 25, 386–388.

Kesten, H. (1958), "Accelerated Stochastic Approximation," *The Annals of Mathematical Statistics*, 29, 41–59.

Kiefer, J. and Wolfowitz, J. (1952), "Stochastic Estimation of the Maximum of a Regression Function," *The Annals of Mathematical Statistics*, 23, 462–466.

Kuk, A. Y. C. (1995), "Asymptotically Unbiased Estimation in Generalized Linear Models With Random Effects," *Journal of the Royal Statistical Society B*, 57, 395–407.

Kuk, A. Y. C. and Cheng, Y. W. (1997), "The Monte Carlo Newton-Raphson Algorithm," *Journal of Statistical Computation and Simulation*, 59, 233–250.

Kushner, H. J. (1987), "Asymptotic Global Behavior for Stochastic Approximation and Diffusions With Slowly Decreasing Noise Effects: Global Minimization Via Monte Carlo," *SIAM Journal on Applied Mathematics*, 47, 169–185.

Laird, N., Lange, N., and Stram, D. (1987), "Maximum Likelihood Computations With Repeated Measures: Application of the EM Algorithm," *Journal of the American Statistical Association*, 82, 97–105.

Lange, K. (1995), "A Gradient Algorithm Locally Equivalent to the EM Algorithm," *Journal of the Royal Statistical Society B*, 57, 425–437.

Langford, I. and Bentham, G. (1997), "A Multilevel Model of sudden Infant Death Syndrome in England and Wales," *Environment and Planning A*, 29, 629–640.

Lee, L.-F. (1992), "On Efficiency of Methods of Simulated Moments and Maximum Simulated Likelihood Estimation of Discrete Response Models," *Econometric Theory*, 8, 518–552.

Lee, L.-F. (1995), "Asymptotic Bias in Simulated Maximum Likelihood Estimation of Discrete Choice Models," *Econometric Theory*, 11, 437–483.

Lee, L.-F. (1998), "Simulated Maximum Likelihood Estimation of Dynamic Discrete Choice Statistical Models: Some Monte Carlo Results," *Journal of Econometrics*, 82, 1–35.

Lee, L.-F. (1999), "Statistical Inference with Simulated Likelihood Functions," *Econometric Theory*, 15(3), 337–360.

Lee, Y. and Nelder, J. A. (1996), "Hierarchical Generalized Linear Models," *Journal of the Royal Statistical Society B*, 58, 619–678.

Levine, R. A. and Casella, G. (2001), "Implementations of the Monte Carlo EM algorithm," Technical report, University of Florida, Department of Statistics.

Liao, J. G. (1998), "Variance Reduction in Gibbs Sampler Using Quasi Random Numbers," *Journal of Computational and Graphical Statistics*, 7, 253–266.

Liesenfeld, R. (1998), "Dynamic Bivariate Mixture Models: Modeling the Behavior of Prices and Trading Volume," *Journal of Business and Economic Statistics*, 16, 101–109.

Lin, X. and Breslow, N. E. (1996), "Bias Correction in Generalized Linear Mixed Models With Multiple Components of Dispersion," *Journal of the American Statistical Association*, 91, 1007–1016.

Littell, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996), *SAS System for Mixed Models*, Cary, NC: SAS Institute.

Liu, C. and Rubin, D. B. (1994), "The ECME Algorithm: A Simple Extension of EM and ECM With Faster Monotone Convergence," *Biometrika*, 81, 633–648.

Liu, C., Rubin, D. B., and Wu, Y. N. (1998), "Parameter Expansion to Accelerate EM: the PX-EM Algorithm," *Biometrika*, 85, 755–770.

Liu, Q. and Pierce, D. A. (1994), "A Note on Gauss-Hermite Quadrature," *Biometrika*, 81, 624–629.

Louis, T. A. (1982), "Finding the Observed Information Matrix when using the EM algorithm," *Journal of the Royal Statistical Society B*, 44, 226–233.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis*, New York: Academic Press.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, New York: Chapman & Hall.

McCulloch, C. E. (1994), "Maximum Likelihood Variance Components Estimation for Binary Data," *Journal of the American Statistical Association*, 89, 330–335.

McCulloch, C. E. (1997), "Maximum Likelihood Algorithms for Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 92, 162–170.

McCulloch, C. E. and Searle, S. (2001), *Generalized, Linear and Mixed Models*, New-York: Wiley.

McKendrick, A. (1926), "Applications of Mathematics to medical problems," *Proceedings of the Edinburgh Mathematical Society*, 44, 98–130.

Meilijson, I. (1989), "A Fast Improvement to the EM Algorithm on Its Own Terms," *Journal of the Royal Statistical Society B*, 51, 127–138.

Meng, X.-L. (1994), "On the Rate of Convergence of the ECM Algorithm," *The Annals of Statistics*, 22, 326–339.

Meng, X.-L. and Rubin, D. B. (1991), "Using EM to Obtain Asymptotic Variance-covariance Matrices: The SEM Algorithm," *Journal of the American Statistical Association*, 86, 899–909.

Meng, X.-L. and Rubin, D. B. (1993), "Maximum Likelihood Estimation Via the ECM Algorithm: A General Framework," *Biometrika*, 80, 267–278.

Moon, C.-G. and Stotsky, J. G. (1993), "Testing the Differences Between the Determinants of Moody's and Standard & Poor's Ratings. An Application of Smooth Simulated Maximum Likelihood Estimation," *Journal of Applied Econometrics*, 8, 51–69.

Morokoff, W. J. and Caflisch, R. E. (1995), "Quasi-Monte Carlo Integration," *Journal of Computational Physics*, 122, 218–230.

Morokoff, W. J. and Caflisch, R. E. (1998), "Quasi-Monte Carlo Simulation of Random Walks in Finance," In *Monte Carlo and quasi-Monte Carlo methods*, pages 340–352, New York: Springer.

Natarajan, R., McCulloch, C., and Kiefer, N. (2000), "A Monte Carlo EM method for estimating Multinomial Probit Models," *Computational Statistics and Data Analysis*, 34, 33–50.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society A*, 135, 370–384.

Neuhaus, J. M., Hauck, W. W., and Kalbfleisch, J. D. (1992), "The Effects of Mixture Distribution Misspecification When Fitting Mixed-effects Logistic Models," *Biometrika*, 79, 755–762.

Niederreiter, H. (1978), "Quasi-Monte Carlo Methods and Pseudo-Random Numbers," *Bulletin of the American Mathematical Society*, 84(6), 957–1041.

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia: SIAM.

Niederreiter, H. and Peart, P. (1986), "Localization of Search in Quasi-Monte Carlo Methods for Global Optimization," *SIAM Journal on Scientific and Statistical Computing*, 7, 660–664.

Oakes, D. (1999), "Direct Calculation of the Information Matrix via the EM Algorithm," *Journal of the Royal Statistical Society B*, 61, 479–482.

Oh, M.-S. and Berger, J. O. (1993), "Integration of Multimodal Functions By Monte Carlo Importance Sampling," *Journal of the American Statistical Association*, 88, 450–456.

Ostland, M. and Yu, B. (1997), "Exploring Quasi Monte Carlo for Marginal Density Approximation," *Statistics and Computing*, 7, 217–228.

Owen, A. B. (1998), "Monte Carlo Extension of Quasi-Monte Carlo," In *1998 Winter Simulation Conference Proceedings*, pages 571–577, New York: Springer.

Owen, A. B. and Zhou, Y. (2000), "Safe and Effective Importance Sampling," *Journal of the American Statistical Association*, 95, 135–143.

Pagès, G. (1992), "Van der Corput Sequences, Kakutani Transforms and One-Dimensional Numerical Integration," *Journal of Computational and Applied Mathematics*, 44, 21–39.

Pan, J.-X. and Thompson, R. (1998), "Quasi-Monte Carlo EM Algorithm for MLEs in Generalized Linear Mixed Models," In *COMPSTAT – Proceedings in Computational Statistics, 13th Symposium*, pages 419– 424, New York: Springer.

Paskov, S. H. (1997), "New Methodologies for Valuing Derivatives," In *Mathematics of Derivative Securities*, pages 545–582, Cambridge: Cambridge University Press.

Paskov, S. H. and Traub, J. (1995), "Faster Valuation of Financial Derivatives," *The Journal of Portfolio Management*, 22, 113–120.

Pinheiro, J. C. and Bates, D. M. (1995), "Approximations to the Log-likelihood Function in the Nonlinear Mixed-effects Model," *Journal of Computational and Graphical Statistics*, 4, 12–35.

Ramamurti, R. (2000), "A multilevel Model of Privatization in Emerging Economies," *Academy of Management Review*, 25, 525–550.

Raudenbush, S., Brennan, R., and Barnett, R. (1995), "A Multivariate Hierarchical Model for studying Psychological Change within Married-Couples," *Journal of Family Psychology*, 9, 161–174.

Raudenbush, S. and Bryk, A. (1986), "A Hierarchical Model for studying School Effects," *Sociology of Education*, 59, 1–17.

Redner, R. A. and Walker, H. F. (1984), "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review*, 26, 195–202.

Robbins, H. and Monro, S. (1951), "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, 22, 400–407.

Robert, C. P. and Casella, G. (1999), *Monte Carlo Statistical Methods*, New York: Springer.

Rubin, D. B. (1991), "EM and Beyond," *Psychometrika*, 56, 241–254.

Ruppert, D. (1985), "A Newton-Raphson Version of the Multivariate Robbins-Monro Procedure," *The Annals of Statistics*, 13, 236–245.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992), *Variance Components*, New York: Wiley.

Shaw, E. (1988), "A Quasirandom Approach to Integration in Bayesian Statistics," *The Annals of Statistics*, 16, 895–914.

Sobol, I. M. (1967), "Distribution of Points in a Cube and Approximate Evaluation of Integrals," *U.S.S.R. Computational Mathematics and Mathematical Physics*, 7, 784–802.

Stein, G. Z. (1988), "Modelling Counts in Biological Populations," *The Mathematical Scientist*, 13, 56–65.

Tang, Q.-Y., L'Ecuyer, P., and Chen, H.-F. (1999), "Asymptotic Efficiency of Perturbation-Analysis-Based Stochastic Approximation with Averaging," *SIAM Journal on Control and Optimization*, 37, 1822–1847.

Tezuka, S. and Fushimi, M. (1992), "A Fast Quasi-Monte Carlo Method Based on the Polynomial Arithmetic Over a Galois Field," *Japanese Journal of Applied Statistics*, 21, 37–48.

Thompson, R. and Meyer, K. (1986), "Estimation of Variance Components: What Is Missing in the EM Algorithm?," *Journal of Statistical Computation and Simulation*, 24, 215–230.

Tierney, L. and Kadane, J. B. (1986), "Accurate Approximations for Posterior Moments and Marginal Densities," *Journal of the American Statistical Association*, 81, 82–86.

Wang, I.-J., Chong, E. K. P., and Kulkarni, S. R. (1997), "Weighted Averaging and Stochastic Approximation," *Mathematics of Control, Signals, and Systems*, 10, 41–60.

Wang, X. and Hickernell, F. J. (2001), "Randomized Halton Sequences," *Mathematical and Computer Modelling*, In press.

Wei, G. C. G. and Tanner, M. A. (1990), "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms," *Journal of the American Statistical Association*, 85, 699–704.

Weil, C. (1970), "Selection of the Valid Number of Sampling Units and Consideration of their Combination in Toxicological Studies Involving Reproduction, Teratogenesis, or Carcinogenesis," *Food and Cosmetic Toxicology*, 8, 177 – 182.

Williams, D. A. (1982), "Extra-binomial Variation in Logistic Linear Models," *Applied Statistics*, 31, 144–148.

Wolfinger, R. and O'Connell, M. (1993), "Generalized Linear Mixed Models: A Pseudo-likelihood Approach," *Journal of Statistical Computation and Simulation*, 48, 233–243.

Wolfowitz, J. (1956), "On Stochastic Approximation," *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, 1, 39– 55.

Woodhouse, G., Yang, M., Goldstein, H., and Rasbash, J. (1996), "Adjusting for Measurement Error in Multilevel Analysis," *Journal of the Royal Statistical Society A*, 159, 201–212.

Wu, C. F. J. (1983), "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics*, 11, 95–103.

Zhang, P. (1996), "Nonparametric Importance Sampling," *Journal of the American Statistical Association*, 91, 1245–1253.

BIOGRAPHICAL SKETCH

Wolfgang Jank was born in Graz, Austria, on March 18, 1970. He moved to Aachen, Germany, at the age of six where he attended high school and university. He graduated from the Technical University of Aachen in May 1996 with a Master of Science degree in mathematics with emphasis on mathematical statistics and a minor in business administration. He satisfied parts of the requirements for his degree at the University of York, England, and at the Université de Montpellier, France, where he studied on a European Community fellowship. It must have been during this time when he developed the desire to spend more of his life in a foreign country.

Thus, Wolfgang decided to become a Gator and moved to Florida in the summer of 1996. While at the University of Florida he worked as a teaching and research assistant, taught undergraduate level classes and worked as a statistical consultant in the Department of Family, Youth and Community Sciences. After graduating from the University of Florida, Wolfgang will move to the Washington, DC, metro area to work as an assistant professor in the Department of Decision and Information Technology in the Robert H. Smith School of Business at the University of Maryland.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

James G. Booth, Chairman
Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Alan Agresti
Distinguished Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

James Hobert
Associate Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Prett Presnell
Associate Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Kirk Hatfield
Associate Professor of Civil and Coastal Engineering

This dissertation was submitted to the Graduate Faculty of the Department of Statistics in the College of Liberal Arts and Sciences and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2001

Dean, Graduate School

STOCHASTIC ESTIMATION METHODS IN GENERAL HIERARCHICAL MODELS

Wolfgang S. Jank
(352) 392–1941
Department of Statistics
Chair: Dr. James G. Booth
Degree: Doctor of Philosophy
Graduation Date: August 2001

In this work we compare different methods to perform maximum likelihood computation for hierarchical models. Hierarchical models are useful tools that enable the researcher to account for a natural grouping or clustering among the observations. One disadvantage of this general class of models is that the likelihood function typically involves an intractable integral, making maximum likelihood estimation complicated. There exist several deterministic approaches to approximate the intractable likelihood function. However, these approaches do not always work well. Stochastic approaches form a popular alternative. Stochastic approaches use simulation to approximate the intractable likelihood integral. They are computer intensive techniques and have become increasingly popular with the development of faster and more powerful computing facilities. In this dissertation we compare five different stochastic estimation methods. We describe each of the five methods, point out limitations and difficulties and perform analytical as well as empirical investigations to compare the efficiency of these methods.